Karlsruhe Institute of Technology

Distributional Tensor Space Model of Natural Language Semantics

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften *(Dr.-Ing.)* von der
Fakultät für Wirtschaftswissenschaften des
Karlsruher Instituts für Technologie (KIT)
genehmigte
DISSERTATION
von

M.Sc. Eugenie Giesbrecht

Tag der mündlichen Prüfung:	28.02.2014
Referent:	Prof. Dr. Rudi Studer
Korreferent:	Prof. Dr. Sebastian Rudolph
Prüfer:	Prof. Dr. Thomas Setzer
Vorsitzender:	Jun.-Prof. Dr. Marten Hillebrand

To my family

Acknowledgements

First and foremost, I would like to thank Prof. Dr. Rudi Studer for giving me the opportunity to do research on this fascinating topic, for his gentle and persuasive guidance, and for the great atmosphere in ”Semantic Karlsruhe”.

I am deeply grateful to Prof. Dr. Sebastian Rudolph for lots of joint brainstorming, encouraging feedback, and his readiness to help at any time day and night.

A special thanks goes also to Marco Baroni and Alessandro Lenci for fruitful discussions and suggestion of new evaluation exercises.

Further, I would like to thank Prof. Dr. Thomas Setzer and Jun.-Prof. Dr. Marten Hillebrand for taking time to review my thesis and attend my PhD defense.

Moreover, I am much obliged to Dr. Andreas Abecker. I very much appreciate his rigorous sense of humour and his unique troubleshooting style which made me learn a lot during our common projects at Forschungszentrum Informatik (FZI). Furthermore, I am thankful to Prof. Dr.-Ing. habil. Catherina Burghart for continuous motivation and support during my work at FZI.

All in one, I am happy to have worked with such outstanding mentors, excellent colleagues and students.

Last but not least, I am deeply grateful to my family and my friends for being in my life and for trust in me.

Abstract

This thesis presents Distributional Tensor Space Model (DTSM) of natural language semantics.

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 of the following two key requirements: (1) sensitivity to structural information such as word order and (2) linguistically justified operations for semantic composition. Especially the problem of compositionality of natural language has been gaining in importance in the scientific community since 2008.

The currently predominant methods are based on matrices, that is 2nd-order tensors. We propose a novel approach that offers a potential of integrating both aspects by employing 3d-order tensors that accounts for order-dependent word contexts and assigns to words characteristic matrices such that semantic composition can be realized in a linguistically and cognitively plausible way.

The DTSM was evaluated on the existing reference datasets as well as on the self-initiated benchmarks that were provided for competitions at workshops co-located with top conferences in computational linguistics. The proposed model achieves state-of-the-art results for important tasks of linguistic semantics by using a relatively small text corpus, without any sophisticated preprocessing and ambitious parameter optimization.

Zusammenfassung

In der Dissertation wird ein distributionelles tensorbasiertes Modell für die Semantik natürlicher Sprache vorgestellt.

Distributionelle Semantik ist einer der erfolgreichsten Formalismen der Sprachsemantik. Dabei wurden mindestens zwei Probleme solcher Modelle bis vor kurzem grötenteils ignoriert: die Wichtigkeit der Wortreihenfolge im Text sowie die Notwendigkeit, Kompositionalität der Sprache linguistisch plausibel abzubilden. Vor allem der letztere Aspekt der Kompositionalität wird erst seit circa 2008 in der Forschungsgemeinschaft als wissenschaftliche Herausforderung zunehmend anerkannt.

Die zurzeit dominierenden Verfahren bauen auf Matrizen, d.h. Tensoren vom Grad 2, auf. In der vorliegenden Arbeit wird ein Modell vorgeschlagen, welches eine Lösung für beide Problempunkte der distributionellen Semantik anbietet. Hierfür wird das Konzept der Tensor-Räume (mit Tensoren vom Grad 3) verwendet, welche es erlauben, sowohl die Reihenfolge der Wörter im Text im Modell zu beachten, als auch eine Matrix-basierte distributionelle Wortrepräsentation zu gewinnen. Matrizen anstelle von Vektoren als Wortrepräsentationsformalismus bietet uns die Möglichkeit, die Kompositionalität der Sprache linguistisch und kognitiv adäquat darzustellen.

Das Modell wurde anhand existierender Referenzdatensätze sowie mit Hilfe der selbst initiierten Benchmarks evaluiert und erzielte Resultate, die dem gegenwärtigen "state of the art" entsprechen; mit dem Vorteil dass der entwickelte Ansatz keine anspruchsvolle Vorverarbeitung erfordert, auf einem kleinen Text-Corpus basiert und in der Lage ist, die meisten Aufgaben der linguistischen Semantik anhand von einmal erzeugten Modellen zu bewältigen, ohne weitere Parameteroptimierung.

Contents

List of Figures **ix**

List of Tables **xi**

1 Introduction **1**

1.1 Research Questions 5

1.2 Thesis Structure 7

1.3 Relevant Publications 8

2 Mathematical Preliminaries **11**

3 Related Work **19**

3.1 Formalization of Semantic Space Models 19

3.2 Review of Semantic Space Models 20

3.2.1 Word-Based Models 20

3.2.2 Randomized methods 24

3.2.3 Dependency-Based Semantic Space Models 28

3.2.4 Distributional Memory Framework 30

3.2.5 Tensor Approaches 32

3.2.6 Summary 33

3.3 Compositionality in Semantic Space Models 35

3.3.1 Mathematical Compositionality Operators 38

3.3.2 Integrating Formal and Distributional Semantics 40

3.4 Conclusion and Outlook 43

4 Distributional Tensor Space Model (DTSM) 45
4.1 Three-Way Model of Distributional Semantics 45
4.2 Compositional Matrix Space Model (CMSM) as a Framework for Language Compositionality 48
4.2.1 Algebraic Plausibility – Structural Operation Properties 49
4.2.2 Neurological Plausibility – Progression of Mental States 50
4.2.3 Psychological Plausibility – Operations on Working Memory . . 50
4.2.4 CMSMs Encode Vector Space Models 51
4.2.4.1 Vector Addition 51
4.2.4.2 Component-wise Multiplication 52
4.2.4.3 Holographic Reduced Representations 52
4.2.4.4 Permutation-based Approaches 53
4.2.5 CMSMs Encode Symbolic Approaches 54
4.2.5.1 Group Theory 54
4.2.5.2 Regular Languages 54
4.2.5.3 The General Case: Matrix Grammars 55
4.2.6 Combination of Different Approaches 59

5 Evaluation Procedure 61
5.1 Datasets 61
5.2 Evaluation Metrics 62
5.3 Computational Resources and Frameworks 63
5.4 Construction of Distributional Tensor Space Model 64
5.5 Tensor Exploitation 65

6 Experimental Part I: Model Evaluation 69
6.1 Free Word Associations Task 70
6.1.1 Procedure 71
6.1.2 Results 71
6.2 Similarity Judgements 73
6.3 Thematic Fit, or Selectional Preferences 75
6.3.1 Procedure 77
6.3.2 Results 78

7 Experimental Part II: Evaluation of Compositionality 81
7.1 Non-Compositionality Detection for Multiword Units 83
7.1.1 Compositional Models 85
7.1.2 Experimental Setup 87
7.1.3 Results 88
7.2 Graded Compositionality 91
7.2.1 Shared Task Dataset Description 92
7.2.2 Dataset Statistics 94
7.2.3 Evaluation Measures 94
7.2.4 Participants of DiSCo and Official Results 95
7.2.5 DTSM Evaluation with DiSCo Dataset 99
7.2.5.1 Procedure 99
7.2.5.2 Parameters of the Model 100
7.2.5.3 Results 101
7.2.6 Summary 110
7.3 Semantic (Non-)Compositionality in Context 113
7.3.1 Procedure 115
7.3.2 Results 116
7.4 Phrasal Semantics 119
7.4.1 Procedure 120
7.4.2 Results 121
7.5 Predicting similarity judgments on transitive sentences 127
7.5.1 Procedure 127
7.5.2 Results 128

8 Summary of Contributions and Outlook 131
8.1 Conclusions from Evaluation 133
8.2 Outlook 135
8.3 Recap of Contributions 136

References 137

List of Figures

2.1 Graphical Representation of a Tensor with $d = 3$ 13

2.2 Visualisation of Tucker Decomposition [Kolda, 2007] 14

2.3 Visualisation of Parafac Decomposition [Kolda, 2007] 15

2.4 Cosine (θ) and Euclidean (d) Similarity Metrics in a Vector Space . . . 16

5.1 X-Slice of Tensor 66

5.2 Y-Slice of Tensor 66

5.3 Z-Slice of Tensor 67

5.4 Intersection of two slices: row vector, or fiber, extraction 67

5.5 Intersection of three slices: finding point values 68

7.1 Two-dimensional word space 85

7.2 Composition operations on MWU in a 2-dimensional word space 87

7.3 Precision-recall diagram 90

7.4 Sample of the data for **V_OBJ: buck trend** [Biemann and Giesbrecht, 2011] 93

7.5 Example of SemEval, Task 5b, Dataset 114

7.6 Visualization of cosine for *addition*, *multiplication* and *MWU* in two dimensions 122

LIST OF FIGURES

List of Tables

1.1 Vector Space Model (VSM) 3
1.2 Word Space Model (WSM) 4
1.3 Word Space Model for ”Peter kicked the ball” 5

4.1 Slice for target word KICK 46
4.2 Slice for target word BALL 46

6.1 Free word associations: Accuracies for *equally distributed* threshold . . . 72
6.2 Free word associations: Accuracies for *linear growing* threshold 72
6.3 Rubenstein and Goodenough (1965): Pearson correlations for DTSM, (recall in parenthesis) 74
6.4 Rubenstein and Goodenough (1965): Pearson correlations for DTSM_{best} with varying minimum triple occurrences 74
6.5 Rubenstein and Goodenough (1965): Pearson correlations for DTSM and state-of-the-art models of similar background 74
6.6 Precision of DTSM on **bicknell.64** dataset for triples with minOcc=5 and modifying agents 79
6.7 Precision of DTSM on **bicknell.64** dataset for triples with minOcc=5 and modifying patients 80

7.1 An example of a word space containing MWU 88
7.2 MWU dataset: similarity values for evaluated compositionality models . 89
7.3 DiSCo English dataset: number of target phrases (with coarse scores) [Biemann and Giesbrecht, 2011] 94
7.4 Participants of DiSCo-2011 Shared Task [Biemann and Giesbrecht, 2011] 96

7.5 Numerical evaluation scores for English @ DiSCo: average point difference and correlation measures; non-significant values are in parentheses 98

7.6 Coarse evaluation scores for submitted results @ DiSCo 98

7.7 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, pmi, $f = 100$ and *add* for different choices of context words 101

7.8 DiSCo dataset: average point difference and correlation measures for DTSM with $min = 5$, pmi, $f = 100$ and *add* for varying context dimensions and context window sizes 103

7.9 DiSCo dataset: coarse evaluation scores for DTSM with $min = 5$, pmi, $f = 100$ and *add* for varying context dimensions and context windows . 103

7.10 DiSCo dataset: average point difference and correlation measures for DTSM with $min = 5$, pmi, $f = 100$, *mult* and varying context window sizes 104

7.11 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, pmi, *add* and $f = 50/100/200$ or without decomposition 104

7.12 DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, pmi, *add* and $f = 50/100/200$ or w/t decomposition . 105

7.13 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 10$, $nn = 13$, pmi, *mult* and $f = 50/100$ or without decomposition 105

7.14 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, $f = 100$, *add* and using *boolean*, *frequency* and *pmi* weighting functions 105

7.15 DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, $f = 100$, *add* and using *boolean*, *frequency* and *pmi* weighting functions 106

7.16 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $f = 100$, pmi and *add* or *mult* 106

7.17 DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $f = 100$, pmi and *add* or *mult* 106

7.18 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 2000$, $f = 100$, $nn = 13$, *pmi*, *add* and using varying minimum triple occurrences size . . . 107

7.19 DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 2000$, $f = 100$, $nn = 13$, *pmi*, *mult* and using varying minimum triple occurrences size . . . 107

7.20 DiSCo: average point difference and correlation measures using $\langle X, Y, Z \rangle$ slices with $dim = 5000$, $min = 5$, $nn = 13$, $f = 50$, *pmi* and different compositionality models (*add*, *mult*, *mixed*) . . . 108

7.21 DiSCo: average point difference and correlation measures for the "coarse" dataset compared to the "numerical dataset"; with $f = 100$ and weighting function PMI . . . 109

7.22 DiSCo dataset: excerpt of phrases where numerical score difference between *gold standard* and *system* is bigger than 40 . . . 110

7.23 Summary of DTSM Evaluation on DiSCo Dataset . . . 111

7.24 DTSM$_{best}$ ($dim = 5000$, $min = 5$, $nn = 13$, $f = 50$, *pmi*) compared to the best DiSCo participating systems: numerical scores and overall correlation . . . 112

7.25 DTSM$_{best}$ compared to DiSCo participating systems: coarse scores . . . 112

7.26 SemEval 2013 Task 5b: Average cosine similarity values for literal and figurative vectors . . . 116

7.27 SemEval 2013 Task 5b: DTSMresults . . . 117

7.28 SemEval 2013 Task 5b: evaluating DTSM for different context choices . 118

7.29 SemEval 2013 Task 5b: DTSM$_{best}$ versus other models . . . 118

7.30 Descriptive statistics for Mitchell and Lapata [2010] dataset: Human performance . . . 119

7.31 Descriptive statistics for DTSM on Mitchell and Lapata (2010) dataset . 121

7.32 Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $nn = 2$, $f = 50$ and varying context dimension choice . . 123

7.33 Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $f = 50$ and varying context window size . . . 123

7.34 Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $nn = 13$ and varying factors size . . . 123

7.35 Spearmans ρ for DTSM ($dim = 2000$, $nn = 13$, $min = 5$), models from Mitchell and Lapata [2010] and human similarity ratings 125

7.36 An example of Grefenstette and Sadrzadeh [2011b] dataset 127

7.37 Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $min = 5$, $nn = 13$, $f = 50$ and varying context dimensions size; insignificant values are in parenthesis (with $p > 0.05$) 129

7.38 Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $min = 5$, $nn = 13$ and varying number of decomposition factors; insignificant values are in parenthesis (with $p > 0.05$) . . . 129

7.39 Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $nn = 13$, $f = 100$ and varying minimum triple occurrences; insignificant values are in parenthesis (with $p > 0.05$) 129

7.40 Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $min = 5$, $nn = 13$, $f = 50$ and varying tensor decomposition methods; insignificant values are in parenthesis (with $p > 0.05$) 130

7.41 DTSM Spearman's correlation compared to state-of-the-art methods [Grefenstette et al., 2013] 130

1

Introduction

Remember "The Hitchhiker's Guide to the Galaxy", where a supercomputer with a profound name "Deep Thought" was built to calculate "the Answer to Life, the Universe, and Everything". It took the computer 7,5 million years to respond with just one word - "42". Seeing the human's dissatisfaction with the answer, the computer wondered suddenly, what the question was actually about. Nowadays, computers need only fractions of seconds to respond to our inquiries, and we turn to search engines to find out any kind of information ranging from business to private questions, be it the newest events in politics, the opinions on the last election campaign, the weather tomorrow or even how to dress for the New Year's party.

The speed of computers has increased immensely and it is not necessary to wait for millions of years to get any information, but the quality of automatically returned information in natural language still leaves much to be desired.

One of the most prominent problems on the way for the computer to pass the "Turing Test"[1] is the problem of natural language representation, i.e., how English, German or any other language in the world can be described in such a way, that a computer can interpret it and the semantics of corresponding languages is correctly reflected in such a representation. Currently it primarily concerns written language, as we are dealing with huge amounts of textual information available in the digital form.

[1] The famous Turing test, proposed by Alan Turing in 1950, is about evaluating machine intelligence by means of parallel communcation with a machine and a human in written natural language. When a human cannot differentiate between a human and a machine answer, then the machine has passed the Turing intelligence test.

1. INTRODUCTION

Vector space models (VSM) [Salton et al., 1975] of meaning are arguably one of the most successful paradigms in computational meaning modeling. VSM embody the distributional hypothesis of meaning, best described by the famous slogan widely attributed to the English linguist John Rupert Firth [1957] that "**a word is known by the company it keeps**".

However, this idea was first mentioned as early as 1884 by Gottlob Frege [Frege, 1884]; and it is known as the **context principle**. Frege suggested that one should "never ask for the meaning of a word in isolation, but only in the context of a sentence".

Following Gottlob Frege, the **context principle** is manifested in early works of Ludwig Wittgenstein, an Austrian-British philosopher and logician, who recapitulates that "an expression has meaning only in a proposition" [Wittgenstein, 1922, 1953].

Almost at the same time, a Russian-American structural linguist and mathematical syntactician Zellig Sabbettai Harris suggests that the degree of semantic similarity between two linguistic expressions A and B is a function of the similarity of the linguistic contexts in which A and B can appear [Harris, 1954].

These ideas experienced revival through vector space models with the development of computer science and especially information retrieval [Salton et al., 1975; Deerwester et al., 1990].

The second birth of distributional semantics in the original linguistic sense was due to the works of Schütze [1992, 1993, 1998] in computational linguistics as well as the work of Landauer and Dumais [1997] in cognitive psychology.

Thereafter, the vector space model and its variations, such as Word Space Models [Schütze, 1993], Hyperspace Analogue to Language [Lund and Burgess, 1996], or Latent Semantic Analysis [Deerwester et al., 1990], have become the mainstream for meaning representation in natural language processing.

Later, the idea was taken up in cognitive science by Peter Gärdenfors [Gärdenfors, 2004], where **conceptual spaces** based on vector space models were suggested as a bridge between formal and connectionist approaches for human concept formation.

Vector Space Model (VSM). The meaning of a word in VSM is defined by contexts in which it occurs in a given text collection. The contexts can be either *local*, e.g., just the word's immediate neighbours or the sentence it occurs in, or *global*, e.g., a paragraph

[Landauer and Dumais, 1997] or a whole document like a Wikipedia article [Gabrilovich and Markovitch, 2007].

Typically, a global, i.e. a bigger, context is used for modeling words' meanings in information retrieval. To do this, a term-document matrix is constructed and the meaning of the terms is defined by the documents they co-occur in. Table 1.1 shows an example of such a space that consists of three documents and 5 key terms.

	document1	**document2**	**document3**
raining	2	0	1
cats	2	2	0
dogs	2	3	0
animals	0	1	0
weather	1	0	1

Table 1.1: Vector Space Model (VSM)

Thus, if you would be looking for information about "cats", you would obtain links to the documents 1 and 2; and if a search engine would offer you related keywords, it may suggest "dogs" as both words co-occur in the same documents.

In computational psychology, Latent Semantic Analysis (LSA) [Deerwester et al., 1990] is an extension of a vector space model that uses a word-by-document matrix, like above, and additionally techniques for dimensionality reduction.

In contrast, research in computational linguistics concentrated mostly on modelling local contexts in that word-by-word matrices are built from text collections. The most famous examples of the latter are *Hyperspace Analogue to Language (HAL)* [Lund and Burgess, 1996] and *Word Space Model (WSM)* [Schütze, 1993].

Here, the meaning of a word is modelled as an n-dimensional vector, where the dimensions are defined by the co-occurring words within a predefined context window (w). Such a context window can be defined, for example, by 5 words to the left and to the right of the target word. Table 1.2 shows an example of such a model for three sentences: "Paul kicked the ball slowly. Peter hit the ball slowly."

Let the context w be equal to 2 words to the left and 2 words to the right. Presuming prior stop words removal, we are left with a vocabulary of 6 words $\langle$*Peter*, *Paul*, *kick*, *hit*, *ball*, *slowly*$\rangle$. Taking into account sentence boundaries, we obtain the following $\langle 6 \times 6 \rangle$ distributional matrix for the above sentences (see Table 1.2).

	Peter	**Paul**	**kick**	**hit**	**ball**	**slowly**
Peter	0	0	0	1	1	0
Paul	0	0	1	0	1	0
kick	**0**	**1**	**0**	**0**	**1**	**1**
hit	**1**	**0**	**0**	**0**	**1**	**1**
ball	1	1	1	1	0	2
slowly	0	0	1	1	2	0

Table 1.2: Word Space Model (WSM)

The matrix shows, e.g., that the word *ball* co-occurs with *slowly* in 2 cases and with *Peter*, *Paul*, *kick* and *hit* once in the given text within a context of 2 words to the left and to the right.

Often, dimensionality reduction techniques, like **singular value decomposition** (SVD), are applied to such matrices as they are very sparse, i.e. the majority of values in such a matrix is zero. SVD is a low-rank approximation of the original vector space matrix. By rank reduction, we cut off the dimensions that do not contribute a lot to the meaning of terms. Some information is lost, but the most important one is preserved and emphasized. Therefore, similar words (*hit* and *kick* in our example in Table 1.2) get closer to each other in vector spaces, although the connections between them may not have been explicitly present in the original data. Thereby second-order, or latent, representations are achieved. This kind of dimensionality reduction has been shown to improve performance in a number of text-based domains [Berry et al., 1999].

Applications. Distributional methods in semantics have proven to be very efficient in tackling a wide range of tasks in natural language processing, e.g., word similarity, synonym identification, or relation extraction, as well as in information retrieval, such as clustering and classification, question answering, query expansion, textual advertisement matching in search engines and so on and so forth (see Turney and Pantel [2010] for a detailed overview).

In spite of their practical viability for a lot of NLP tasks, it is unclear, **to what extent a semantic space model in its current form can serve as a model of meaning**.

1.1 Research Questions

In spite of a quick propagation and success of vector space models in many applied areas, it has been long recognized that these models are too weak to represent natural language to a satisfactory extent.

The main question is therefore whether the currently predominating matrix-based semantic space model with vector-based word meaning representation is appropriate as a model of natural language semantics. In the following, we describe three issues that make us believe that there is a need for novel paradigms.

Word Order. With VSM, the assumption is usually made that word co-occurrence is essentially independent from the word order; and all the co-occurrence information is fed into one vector per word. The following example shows, why it is inadequate.

Suppose, our background knowledge corpus consists of one sentence: *Peter kicked the ball.* Assuming prior "stop words" removal, such as *the* or the *dot*, lemmatization of the words as well as a context window of size three, i.e. one word to the left and one word to the right of the target word, Table 1.3 shows the resulting word space model for this sentence.

	Peter	**kick**	**ball**
Peter	0	1	0
kick	1	0	1
ball	0	1	0

Table 1.3: Word Space Model for "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 only be *kicked* by somebody but not *kick* itself; in case of *Peter*, both ways of interpretation should be possible.

The Problem of Compositionality. The next big challenge of distributional semantics is that it has been predominantly used for meaning representation of single words. The question of representing meaning of more complex language units, such as phrases or sentences, has been until recently ignored by the community. Currently there is a quickly emerging and spreading interest in exactly this topic.

In contrast, symbolic or logical approaches to semantics, following the traditions of Montague's semantics and known in the linguistics community as *compositional semantics* [Dowty et al., 1981], were concerned prevalently with composition of individual units into sentences and not with the meaning of those individual units. Within this logical tradition of semantics, the meaning of a sentence "Peter kicked the ball" could be represented as: $kick(Peter, ball)$ that can be obtained by the composition of the constituents $\lambda x \lambda y.kick(x, y)$, $Peter$ and $ball$ extracted word-wise from the original sentence. This expression can be interpreted in the following way: there is some x and some y such that x *kicked* y and in this case x is Peter and y is the ball.

Such constructions are good at conveying the structural or grammatical properties of language, but unfortunately they do not tell us anything about the meaning of individual units. We still may have no idea though, what a *ball* is, or how *kicked* is different from *caught.* It is just assumed that there is a referent in the external world or in the speaker's mind. Lexicon is much more empirical than grammar; and it is therefore harder to formalize [Widdows, 2008]. According to Jones and Sinclair [1974], "one of the troubles with studying lexis was making a start somewhere"', and a brilliant start was made by vector space models.

It just took a while until it became obvious, that distributional semantics need not be fixed only on the lexical word meaning; and that there should be means of modelling composition of words into phrases and sentences within this paradigm.

Word Representation. Last but not least, current paradigms seem to be insufficient even for word meaning representation. More and more researchers come up with the ideas to express word semantics, e.g., by several vectors instead of one.

Hence, we've identified three critical points of meaning representation with vector space models so far:

1. (non-)sensitivity to structural information such as word order;

2. lack of linguistically justified operations for semantic composition and

3. word meaning representation by means of a single vector.

Making use of word space models, a novel paradigm that is called Distributional Tensor Space Model for representing meaning by **introducing a third dimension** into traditional matrix-based vector spaces is proposed. The latter allows us to **integrate word order information** without extra effort and to **assign to words characteristic matrices** such that **semantic composition** can be later realized in a natural way **via matrix multiplication and partially matrix addition**.

Currently predominating distributional semantics models in computational linguistics are based on two-way tensors, i.e. matrices. Most information can be conveyed, however, by three elements. There are a number of theories that confirm this intuition. **Semantic web** is one of those, where the whole modelling is based on triples of information. Therefore, we believe that a three-way tensor should be sufficient for modelling most of straightforward information. However, we do not restrict the tensor model per definition to three dimensions. It is just for the purpose of the current work and due to certain computational restrictions that we make use of only three dimensions.

1.2 Thesis Structure

After we have identified the main challenges of the current distributional semantics paradigm, we proceed in the following way.

Chapter 2 gives a brief introduction to some aspects of linear algebra that are needed to understand the suggested model. In particular, the mathematical concepts of *vectors*, *matrices*, *tensors*, *linear mappings*, *permutations*, *tensor decomposition methods* as well as *similarity measures* in vector spaces are introduced.

Chapter 3 introduces a formalization of semantic space models as defined by Lowe [2001]. Further, it offers a thorough review of the most influential models that we group according to the way that is used to extract context dimensions. In order to avoid the confusion between different definitions and their connotations, we will use a generic term **semantic space models** as a superordinate concept, as it is used by Lowe [2001]. Last but not least, present approaches to the problem of compositionality within the distributional semantics paradigm are discussed.

Motivated by the ideas of distributional semantics and the mathematics behind it, we propose a novel Distributional Tensor Space Model in Chapter 4. We postulate it in terms of the formalism suggested by Lowe [2001] and show the theoretical and practical advantages of our model. Further, a novel type of generic compositional models based on matrix multiplication, called Compositional Matrix Space Model, is introduced. We show its algebraic, linguistic and neurological plausibility as natural language compositionality model. Moreover, we prove that it subsumes most linear-algebra-based operations that have been proposed to model composition in language models as well as formal approaches.

Chapter 5 gives an overview of the datasets that we use, the evaluation metrics as well as the utilised computational resources and corpora. The procedure of tensor construction and deployment is also described in this chapter.

Chapters 6 and 7 report detailed evaluation results for the proposed model in respect to the two aspects of meaning modelling: the word meaning per se and the construction of compositional meaning for phrases and simple sentences. The model is evaluated on a number of the standard benchmarks in distributional semantics as well as on additional self-constructed resources that have been missing and that will hopefully become benchmarks for the community in the future.

Finally, we recap our contributions, summarize findings as well as give an outlook for future research in Chapter 8.

1.3 Relevant Publications

The research underlying this thesis was published in a number of conference publications. Moreover, it was substantially motivated by our previous work in Katz and Giesbrecht [2006].

The material of Chapter 7.1 was published in Giesbrecht [2009]. Giesbrecht [2010] introduces the proposed model which forms the basis of Chapter 4.1 as well as describes the experiment on *free word associations* which is the subject matter of Chapter 6.1.

Most of the material in Chapter 4.2 on Compositional Matrix Space Model is the topic of Rudolph and Giesbrecht [2010].

Furthermore, two international competitions were initiated and partially co-organized at the top conferences in computational linguistics; both resulted in datasets that are used currently by the researchers interested in compositionality models as well as by those concerned with the computational models for idiomatic language.

The first dataset for graded compositionality of phrases was offered at the self-organized Distributional Semantics and Compositionality (DiSCo-2011) workshop, co-located with the ACL conference. The dataset, the participating systems as well the results of evaluation of Distributional Tensor Space Model on this dataset is the major topic of Chapter 7.2.

The second dataset was constructed as part of SemEval-2013 competition. It determines the content of Chapter 7.3.

2

Mathematical Preliminaries

In this section, we recap some aspects of linear algebra to the extent needed to grasp the suggested model that is described in detail in Chapter 4 as well as the related work in Chapter 3.

For a more thorough treatise we refer the reader to a linear algebra textbook such as Strang [1993].

Vectors. Given a natural number n, an n-dimensional vector $\mathbf{v}$ over the reals can be seen as a list (or tuple) containing n real numbers $r_1, \ldots, r_n \in \mathbb{R}$, written $\mathbf{v} = (r_1 \quad r_2 \quad \cdots \quad r_n)$. Vectors will be denoted by lowercase bold font letters and we will use the notation $\mathbf{v}(i)$ to refer to the ith entry of vector $\mathbf{v}$. As usual, we write $\mathbb{R}^n$ to denote the set of all n-dimensional vectors with real entries. Vectors can be added entry-wise, i.e., $(r_1 \quad \cdots \quad r_n) + (r'_1 \quad \cdots \quad r'_n) = (r_1{+}r'_1 \quad \cdots \quad r_n{+}r'_n)$. Likewise, the entry-wise product (also known as Hadamard product) is defined by $(r_1 \quad \cdots \quad r_n) \odot (r'_1 \quad \cdots \quad r'_n) = (r_1 \cdot r'_1 \quad \cdots \quad r_n \cdot r'_n)$.

Matrices. Given two real numbers n, m, an $n \times m$ matrix over the reals is an array of real numbers with n rows and m columns. We will use capital letters to denote matrices and, given a matrix M we will write $M(i, j)$ to refer to the entry in the ith row and the jth column:

$$M = \begin{pmatrix} M(1,1) & M(1,2) \cdots & M(1,j) \cdots & M(1,m) \\ M(2,1) & M(2,2) & & \vdots \\ \vdots & & & \vdots \\ M(i,1) & & M(i,j) & \vdots \\ \vdots & & & \vdots \\ M(n,1) & M(1,2) \cdots & \cdots & \cdots M(n,m) \end{pmatrix}$$

The set of all $n \times m$ matrices with real number entries is denoted by $\mathbb{R}^{n \times m}$. Obviously, m-dimensional vectors can be seen as $1 \times m$ matrices. A matrix can be *transposed* by exchanging columns and rows: given the $n \times m$ matrix M, its transposed version M^T is a $m \times n$ matrix defined by $M^T(i,j) = M(j,i)$.

Linear Mappings. Beyond being merely array-like data structures, matrices correspond to certain type of functions, so called *linear mappings*, having vectors as in- and output. More precisely, an $n \times m$ matrix M applied to an m-dimensional vector $\mathbf{v}$ yields an n-dimensional vector $\mathbf{v}'$ (written: $\mathbf{v}M = \mathbf{v}'$) according to

$$\mathbf{v}'(i) = \sum_{j=1}^{m} \mathbf{v}(j) \cdot M(i,j) \tag{2.1}$$

Linear mappings can be concatenated, giving rise to the notion of standard matrix multiplication: we write $M_1 M_2$ to denote the matrix that corresponds to the linear mapping defined by applying first M_1 and then M_2.

Permutations. Given a natural number n, a *permutation* on $\{1 \ldots n\}$ is a bijection (i.e., a mapping that is one-to-one and onto) $\Phi : \{1 \ldots n\} \to \{1 \ldots n\}$. A permutation can be seen as a "reordering scheme" on a list with n elements: the element at position i will get the new position $\Phi(i)$ in the reordered list. Likewise, a permutation can be applied to a vector resulting in a rearrangement of the entries. We write Φ^n to denote the permutation corresponding to the n-fold application of Φ and Φ^{-1} to denote the permutation that "undoes" Φ.

Given a permutation Φ, the corresponding *permutation matrix* M_Φ is defined by

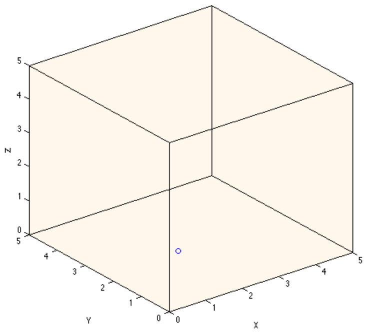

Figure 2.1: Graphical Representation of a Tensor with $d = 3$

$$M_\Phi(i,j) = \begin{cases} 1 \text{ if } \Phi(j) = i, \\ 0 \text{ otherwise.} \end{cases} \tag{2.2}$$

Then, obviously permuting a vector according to Φ can be expressed in terms of matrix multiplication as well as we obtain for any vector $\mathbf{v} \in \mathbb{R}^n$:

$$\Phi(\mathbf{v}) = \mathbf{v} M_\Phi \tag{2.3}$$

Likewise, iterated application (Φ^n) and the inverses Φ^{-n} carry over naturally to the corresponding notions in matrices.

Tensors. 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 $T : \{1, \ldots, n_1\} \times \ldots \times \{1, \ldots, n_d\} \to \mathbb{R}$, 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 $d = 3$ which can be represented graphically as a cube (cf. Figure 2.1).

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

Tensor Decompositions. Several tensor decomposition algorithms have been suggested for dimensionality reduction in three dimensions. Tucker [Tucker, 1966] and CANDECOMP/PARAFAC [Harshman, 1970; Carroll and Chang, 1970] models are the most influential ones. The latter was suggested twice at the same time and received different names by its two proposers emphasizing varying features of the model: *canonical decomposition (CANDECOMP)* [Carroll and Chang, 1970] and *parallel factor analysis (PARAFAC)* [Harshman, 1970]. Therefore, it obtained a double name - the CANDECOMP/PARAFAC (CP) model [Kiers, 2000].

Given an $n_1 \times n_2 \times n_3$ tensor T, its (three-way) **Tucker decomposition** 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:

$$T(i,j,k) = \sum_{r=1}^{m_1} \sum_{s=1}^{m_2} \sum_{t=1}^{m_3} G(r,s,t) \cdot A(i,r) \cdot B(j,s) \cdot C(k,t) + E(i,j,k). \tag{2.4}$$

E(i,j,k) denotes error. Figure 2.2 demonstrates a visualisation of Tucker decomposition.

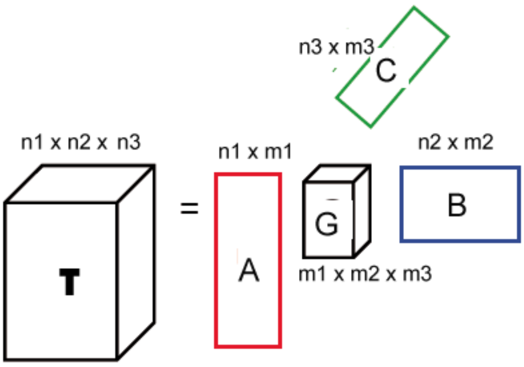

Figure 2.2: Visualisation of Tucker Decomposition [Kolda, 2007]

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, that is, where the error E becomes zero. Rather one may attempt to approximate T as good 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

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

Given an $n_1 \times n_2 \times n_3$ tensor T, its (three-way) **Parafac model** for given natural number m consists of three matrices A, B, and C of formats $n_1 \times m$, $n_2 \times m$, and $n_3 \times m$, respectively and a diagonal core $\lambda(r)$, such that:

$$T(i,j,k) = \sum_{r=1}^{m} \lambda(r) A(i,r) \cdot B(j,r) \cdot C(k,r) + E(i,j,k). \tag{2.5}$$

Consequently, all component matrices have the same number of columns (m). $\lambda(r)$ is a diagonal core which is in this case a vector (not a $3d$ tensor as in the Tucker model) and $E(i,j,k)$ stands for residual error, similarly to Tucker. A visualization of CP model is shown in Figure 2.3.

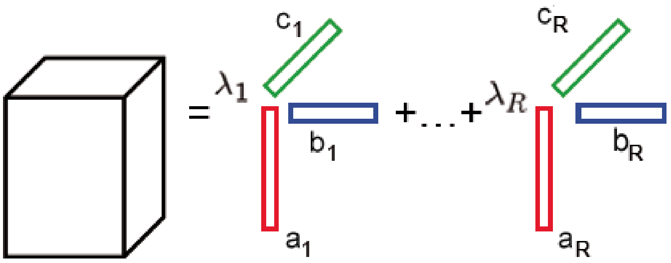

Figure 2.3: Visualisation of Parafac Decomposition [Kolda, 2007]

Non-negative tensor factorization (NTF) [Lee and Seung, 2000] is a relatively recently suggested method for dealing with multi-way data. NTF is a generalization of non-negative matrix factorization (NMF). Formally, it is an extension of the CANDECOMP/PARAFAC model with non-negative constraints on the factors. The version of NTF that is employed in this thesis uses multiplicative updates from the NMF algorithm of Lee and Seung [2000].

In fact, the above described ways of approximating a tensor are called **dimensionality reduction**.

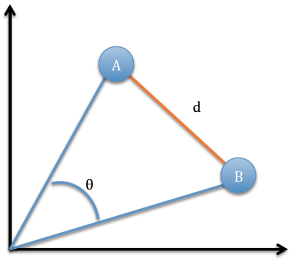

Figure 2.4: Cosine (θ) and Euclidean (d) Similarity Metrics in a Vector Space

Number of Factors. When speaking about dimensionality reduction, one of the most prickling questions is what number of dimensions, or factors, is reasonable to reduce to. Usually the rank of the matrix or the tensor is considered to be equal to the optimal number of factors.

The rank of a matrix is the number of linearly independent row or column vectors of the matrix. The rank of the tensor T is the least number n of rank 1 tensors, the sum of which results in T.

The number of factors may have a decisive role in factor analysis and in the interpretation of the results [Harshman, 1970]. A number of ways have been suggested to determine the best approximation rank mathematically. However, this fact is mostly ignored by non-mathematicians. Usually it is tested empirically what kind of model performs best. We will follow the same procedure in this work and leave the mathematical justification for the best number of factors for future work.

Similarity Measures in Vector Spaces A wide range of similarity measures can be used to measure the similarity between two elements of a vector space[1]. Two most popular of them are cosine similarity and Euclidean distance (cf. Figure 2.4).

Cosine corresponds for normalized unit vectors to a scalar product of those [Manning and Schütze, 1999] (see Equation 2.6).

$$cos(\overrightarrow{x}, \overrightarrow{y}) = \overrightarrow{x} \cdot \overrightarrow{y} \tag{2.6}$$

A normalized vector has a unit length of 1 and is defined as:

[1] See Lee [1999] for an overview.

$$|\overrightarrow{x}| = \sqrt{\sum_{i=1}^{n} x_i^2} = 1 \tag{2.7}$$

In this metric, two expressions are taken to be unrelated if their meaning vectors are orthogonal (the cosine is 0) and synonymous if their vectors are parallel (the cosine is 1).

Euclidean distance is the distance between two vectors that measures how far they are in a vector space from each other:

$$|\overrightarrow{x} - \overrightarrow{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.8}$$

The lower the distance, the higher the similarity.

In spite of apparent differences between the two metrics, they result in similar similarity rankings when applied to high dimensional and normalized vectors [Manning and Schütze, 1999; Qian et al., 2004].

Manning and Schütze [1999, cf. 8.44] show why it happens:

$$(|\overrightarrow{x} - \overrightarrow{y}|)^2 = \sum_{i=1}^{n} (x_i - y_i)^2 = \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 = 1 - 2\sum_{i=1}^{n} x_i y_i + 1 = 2(1 - \overrightarrow{x} \cdot \overrightarrow{y})$$

We choose to use the cosine similarity metrics for this work.

Cosine measure can also be used for the comparison of two matrices or two three-way tensors.

Cosine between matrices M and M' of size $n1 \times n2$ is defined by

$$\frac{\sum_{r=1}^{n_1} \sum_{s=1}^{n_2} M(r,s) \cdot M'(r,s)}{\|M\| \cdot \|M'\|} \tag{2.9}$$

and can take values between -1 and 1.

Similarly, **cosine between three-way tensors** T and T' is defined by

$$\frac{\sum_{r=1}^{n_1} \sum_{s=1}^{n_2} \sum_{t=1}^{n_3} T(r,s,t) \cdot T'(r,s,t)}{\|T\| \cdot \|T'\|} \tag{2.10}$$

Compositional Operations on Matrices. Furthermore, we shortly recap the notion of matrix multiplication and matrix addition.

Given a $n_1 \times n_2$ matrix A and an $n_2 \times n_3$ matrix B, the **matrix product** AB is defined as the $n_1 \times n_3$ matrix C with

$$C(i,k) = \sum_{r=1}^{n_2} A(i,r) \cdot B(r,k). \tag{2.11}$$

Note that the matrix product is associative ($(AB)C = A(BC)$) but not commutative ($AB = BA$ does not hold in general, i.e., the order matters), hinting at matrix multiplication as a suitable candidate for representing compositional semantics of subsequent words.

Matrix addition is defined in straightforward way, by adding entries with the same indices:

$$C(i,k) = A(i,k) + B(i,k) \tag{2.12}$$

Matrix addition is, similar to addition for real numbers and vectors, associative and commutative. However, we decide to evaluate this operation on matrices, as vector addition turned out to achieve very good results in a number of computational tasks in spite of its counter-intuitive way of handling composition. A matrix holds in any case more information than vectors, so that matrix addition may still turn out to be a valuable operation.

3

Related Work

A number of vector space models has been suggested since the 1990s. In order to avoid confusion between different definitions and their connotations, we will use a generic term **semantic space models**, similarly to Lowe [2001], as a superordinate concept.

Semantic space models are, from the perspective of distributional semantics, models of distributional behaviour of words. Meaning is modelled here as an n-dimensional vector, derived from word co-occurrence counts for the expression in question.

In the following, we briefly review the most influential works on semantic space models and demonstrate the differences between those on a simple example sentence:

$$\textbf{The minister copied from his supervisor.} \tag{3.0.1}$$

We start with the formalism for semantic models suggested by Lowe [2001] and then proceed by mapping of the existing approaches, as far as possible, to this formalism as well as by demonstrating the transformation of the example sentence 3.0.1 into semantic space models according to the corresponding algorithms.

3.1 Formalization of Semantic Space Models

A semantic space model has been formalized by Lowe [2001] as a quadruple $\langle B, A, S, M\rangle$ where:

B: defines a set of **basis, or context, elements**;

B can be interpreted in at least two ways: as a concrete list of context elements, or as a definition of the way how these context elements are determined. In the following, we will use the second understanding.

A: is a functional mapping of co-occurrence counts between basis elements and target words in the language that need to be represented;

A is often named as a **weighting function**. A can be defined by pure frequency co-occurrences or by lexical association measures[1].

S: is a similarity measure in a vector space;

Typically either Euclidean distance or cosine similarity is used.

M: is a possible transformation of semantic space, e.g., by reducing its dimensionality. M can also be just an identity mapping.

Some researchers leave this transformation out; the others make use of singular value decomposition or other dimensionality reduction techniques.

3.2 Review of Semantic Space Models

3.2.1 Word-Based Models

Word Space Model (WSM). A **word space model** was suggested in Schütze [1993, 1998] as analogue to a vector space model in information retrieval [Salton et al., 1975; Salton and McGill, 1983].

A Word Space is a symmetrical vector space where the dimensions, or contexts, are defined by the co-occurring words within a certain window, for example, 25 words to the left and to the right of the word [Schütze, 1998]

To formalize WSM of, for example, Schütze [1998] in terms of the above described framework of Lowe [2001]:

B: is defined by 25 words to the left and to the right, of which only a certain number (e.g., 1000) most frequent neighbours are used as context words;

Thus, a 1000 × 1000 matrix is constructed.

[1] Lexical association measures are a variety of statistical measures for identifying lexical associations between words, that range from pure frequency counts to information theoretic measures and statistical significance tests [Evert and Krenn, 2001].

A: is determined either just by word frequencies or is smoothed by log inverse document frequency used in information retrieval[1];

S: is cosine similarity;

M: SVD transformation is applied to the semantic space.

For our example (3.0.1), taking two words to the left and to the right as context resulting in a window of size 5 together with a target word, we will get the following matrix:

$$\begin{pmatrix} \begin{array}{ccccccc} & the & minister & copied & from & his & supervisor \\ the & 0 & 1 & 1 & 0 & 0 & 0 \\ minister & 1 & 0 & 1 & 1 & 0 & 0 \\ copied & 1 & 1 & 0 & 1 & 1 & 0 \\ from & 0 & 1 & 1 & 0 & 1 & 1 \\ his & 0 & 0 & 1 & 1 & 0 & 1 \\ supervisor & 0 & 0 & 0 & 1 & 1 & 0 \end{array} \end{pmatrix} \qquad (3.2.1)$$

Hyperspace Analogue to Language (HAL). HAL model was suggested in Lund and Burgess [1996].

In HAL, a semantic space model is a matrix where the rows contain left neighbours and the columns contain right neighbours of the word. The weight in the corresponding matrix cell is defined by the distance between the words, the closer the words the bigger is the weight. Thus, for our example sentence ((3.0.1)) *minister* and *copied* are immediate neighbours, so this connection will get a maximum weight of 5. The general formula of the weight calculation for the word at position `X` and its neighbouring word at position `X-Y` is the following: $window\_size - (X - Y) + 1$.

The HAL model for our example (3.0.1), with a window of size 5 where four neighbours before the target word define the context, would result in the following matrix:

[1] $a_i = log(N/n_i)$ where n_i is the number of documents that a word occurs in and N is the total number of documents

$$\left(\begin{array}{cccccccc}
 & the & minister & copied & from & his & supervisor \\
the & 0 & 0 & 0 & 0 & 0 & 0 \\
minister & 4 & 0 & 0 & 0 & 0 & 0 \\
copied & 3 & 4 & 0 & 0 & 0 & 0 \\
from & 2 & 3 & 4 & 0 & 0 & 0 \\
his & 1 & 2 & 3 & 4 & 0 & 0 \\
supervisor & 0 & 1 & 2 & 3 & 4 & 0
\end{array}\right) \tag{3.2.2}$$

In order to get a vector for the word *copied*, the row- and the column-vectors for *copied* are combined into one vector: $(4, 5, 0, 0, 0, 0; 0, 0, 0, 0, 5, 4, 3)$.

HAL in terms of the formalism of Lowe [2001] is thus defined in the following way:

B: is specified by a context window of 10 words (in the original paper), whereas rows contain only left neighbours and columns consist of the right contexts;

Lund and Burgess [1996] make use of *Usenet news groups corpus* containing 160 million words, of which they used 70.000 words that occur at least 50 times in the corpus.

A: is defined by the distance between the words;

S: is Minkowski similarity measure, e.g., Euclidean distance;

M: dimensionality reduction is achieved by variance.

Words with low variance are discarded. 100 to 200 most variant vector elements have been used for the experiments in the original paper of Lund and Burgess [1996].

Correlated Occurrence Analogue to Lexical Semantics (COALS). Rohde et al. [2006] suggested COALS that was basically an extension of WSM and HAL models. The matrix is built similarly to WSM, i.e. both left and right contexts of the word are used, but the words' co-occurrences are weighted according to distance as in HAL but also smoothed by Pearson correlation.

Thus, the initial matrix in COALS, before smoothing, for our example sentence (3.0.1) would look like this:

$$\begin{pmatrix} \begin{array}{ccccccc} & the & minister & copied & from & his & supervisor \\ the & 0 & 2 & 1 & 0 & 0 & 0 \\ minister & 2 & 0 & 2 & 1 & 0 & 0 \\ copied & 1 & 2 & 0 & 2 & 1 & 0 \\ from & 0 & 1 & 2 & 0 & 2 & 1 \\ his & 0 & 0 & 1 & 2 & 0 & 2 \\ supervisor & 0 & 0 & 0 & 1 & 2 & 0 \end{array} \end{pmatrix} \tag{3.2.3}$$

The corresponding formal semantic space quadruple in terms of Lowe [2001] is the following:

B: is specified by a context window of 4 words using both left and right contexts, similar to WSM;

Usually, only the most frequent open class words are used for the matrix construction (14000 in the original paper).

A: is defined by the distance between the words;

The closer the words are located to each other, the bigger weight they receive.

S: is normalized by correlation between the vectors as a measure of semantic similarity between two words.

The counts in the matrix are converted into Pearson correlations and the negative values are set to zero while the positive ones are squared.

M: optionally, dimensionality reduction by means of SVD can be done.

In this case, dimensionality reduction produces better results.

3.2.2 Randomized methods

Semantic space models have been mostly hampered by the problems of size and sparsity. In order to avoid those, usually dimensionality reduction techniques, like SVD, have been suggested. However, SVD has turned out to be not feasible for huge text collections in many cases. A further problem with SVD is that it is computed once after an initial matrix is constructed. After that, it is not quite trivial to add new data to the model without re-computing SVD from scratch and without losing the quality of approximation for the new information [Sahlgren, 2005].

Random Indexing. In order to avoid a-priori construction of huge matrices with the following computationally expensive dimensionality reduction, Kanerva [1988] and Kanerva, Kristoferson, and Holst [2000] suggest a technique of Random Indexing for the construction of word space models.

Random Indexing consists basically of two steps:

1. An index vector containing randomly distributed values of (1, -1, 0) of length D is assigned to every context (or basis) word or document. D defines a desired dimensionality of context, i.e. the number of columns in the matrix.

   Let all the words in our example sentence (3.0.1) get randomly assigned 3-dimensional index vectors:

`the`	[0 0 0]
`minister`	[1 0 1]
`copied`	[0 1 0]
`from`	[0 0 -1]
`his`	[1 0 0]
`supervisor`	[1 0 -1]

2. Assuming the window size of, i.e. 2 words before and 2 words after the target word, we process the text and add up the index vectors of co-occurring words to the target word.

   The latter results in the following matrix:

`the`	[1 1 1]
`minister`	[1 1 0]
`copied`	[2 1 0]
`from`	[3 1 -1]
`his`	[2 1 -2]
`supervisor`	[2 0 -2]

Thereby, we achieve an incremental and scalable way to construct a semantic space.

Incremental Semantic Analysis (ISA). Baroni et al. [2007] suggested a new way of combination for word index vectors in random indexing, in order to integrate the learning effect over the time. They call it **Incremental Semantic Analysis**. In contrast to the original **Random Indexing**, where the target word's vector is incremented by the same non-changing index vector of the context word every time they co-occur, ISA also considers **distributional histories** of context words. Every time a target word t co-occurs with a context word c, its *distributional history vector* (h_t) is updated as follows: $h_t + = i \times (m_c h_c + (1 - m_c) s_c)$ where s_c is the *signature*, or *index*, vector of the context word; i is the impact rate; m_c is the factor determining the influence of the distributional history of the context word on the target word; it is realized in ISA as a function of the frequency - the more frequently a word occurs, the less informative it is.

By capturing the history of distributions, ISA is supposed to reproduce second order effects like SVD, e.g., `car` and `automobile` may act as similar context words with time as they are likely to have similar distributions in texts over the time.

BEAGLE. In BEAGLE (Bound Encoding of the Aggregate Language Environment), proposed by Jones and Mewhort [2007], words are also represented as D-dimensional random vectors, but they differentiate between two aspects of word meaning representation: context information about word co-occurrence and word order information. The former is aggregated through vector addition and the latter through vector convolution when processing text sentence-by-sentence.

Every word gets assigned a so-called random **environmental** vector e_i, initiated at random from a Gaussian distribution with $\mu = 00$ and $\sigma = 1/\sqrt{D}$ with D being vector dimensionality. Similarly to **index vectors** in Random Indexing (cf. Sahlgren [2005]), these environmental vectors do not change over time while the word's **memory vector** m_i, like **distributional history vectors** in Baroni et al. [2007], is updated every time a word is encountered. The `memory vector` consists of the word's context vector c_i, which is the sum of the environmental vectors of the other words it co-occurs with in a given sentence, and the order vector o_i.

Coming back to our example (3.0.1): we randomly initialize 3-dimensional environmental vectors from a Gaussian distribution with mean 0.0 and standard deviation $1/\sqrt{3} = 0.577$:

`the`	[0.43 0.59 0.15]
`minister`	**[0.13 -0.49 0.78]**
`copied`	*[-0.05 -0.034 0.53]*
`from`	[-0.12 0.69 -0.23]
`his`	[-0.69 0.5 -0.530]
`supervisor`	[**0.27 -0.07 0.66**]

If we ignore stop words *the*, *from* and *his*, the context vector for `copied` c_{copied} will be equal to the sum of environment vectors for `minister` $e_{minister}$ and `supervisor` $e_{supervisor}$:

$c_{copied} = c_{copied} + [0.40 \quad -0.56 \quad 1.44]$

The order vector is the sum of all directional circular convolutions ($\circledast$) for all n-grams in the sentences containing the target word, though the size of n-grams is usually restricted due to computational reasons. In the paper, the maximal number of words' neighbours was limited to seven. Furthermore, no stop word list is used in this case as function words are important for syntax. By means of example, consider reproducing the order information for the word *minister* in 3.0.1, where Φ denotes the position of the target word.

Bigrams	$minister_1 = e_{the} \circledast \Phi$
	$minister_2 = \Phi \circledast e_{copied}$
Trigrams	$minister_3 = e_{the} \circledast \Phi \circledast e_{copied}$
	$minister_4 = \Phi \circledast e_{copied} \circledast e_{from}$
Quadgrams	$minister_5 = e_{the} \circledast \Phi \circledast e_{copied} \circledast e_{from}$
	$minister_6 = \Phi \circledast e_{copied} \circledast e_{from} \circledast e_{his}$
Tetragrams	$minister_7 = e_{the} \circledast \Phi \circledast e_{copied} \circledast e_{from} \circledast e_{his}$
	$minister_8 = \Phi \circledast e_{copied} \circledast e_{from} \circledast e_{his} \circledast e_{supervisor}$
Pentagram	$minister_9 = e_{the} \circledast \Phi \circledast e_{copied} \circledast e_{from} \circledast e_{his} \circledast e_{supervisor}$

Φ is initiated in the same way as the word's environmental vector and it is fix.

The **order vector** for *minister* $o_{minister}$ is then the sum of all above $\Sigma_{j=1}^{j=n} minister_j$.

$o_{minister} = minister_1 + minister_2 + minister_3 + minister_4 + minister_5 + minister_6 + minister_7 + minister_8 + minister_9$

The final combined memory vector is then the sum of c_i and o_i.

Word Order by Permutation. Sahlgren et al. [2008] propose a computationally lighter alternative to convolution and suggest to incorporate word order information into context vectors by means of permutation.

He ignores sentence boundaries and uses a context window of two neighbours to the left and to the right. Context vectors here are built in the same way as in BEAGLE. Thus, for our example (3.0.1), a context vector for *copied*, assuming the elimination of function words, would be: $c_{copied} = 0 + minister + 0 + 0 + 0$.

Word order can be encoded in two different ways:

1. by differentiating between the preceding (using the inverse permutation Π^{-1}) and the following words (Π);

   Such vectors are called **direct vectors**. The latter correspond to the direction-sensitive HAL representations.

2. by permuting the vector n times (Π^n) depending on the distance between the target and the basis word.

   The order information for the word *copied* in our example, with window size equal to the whole sentence, would be encoded like this:

   $o_{copied} = (\Pi^{-2} the) + (\Pi^{-1} minister) + 0 + (\Pi from) + (\Pi^2 his) + (\Pi^3 supervisor)$.

   These vectors are called **order vectors**.

   Unlike Jones and Mewhort [2007], Sahlgren et al. [2008] ignore stop words, or words with frequency more than 15000 occurrences, not only for construction of **context vectors** but also for **order** and **direction vectors** as this improves drastically the results.

Permutations are computationally less expensive than convolution and can be used with any kind of random vectors including the ones of Jones and Mewhort [2007].

All in one, random indexing is a good alternative to heavy SVD-based methods in cases, where the speed of processing is more important than a slight loss of accuracy.

3.2.3 Dependency-Based Semantic Space Models

In contrast to word-based approaches, there are models that incorporate syntax information into the process of semantic space construction.

Grefenstette [1994] and Lin [1998] use syntactic parsers - SEXTANT and MINIPAR [Lin, 1993, 1994] respectively - to extract dependency triples from text. A dependency triple contains two words and a grammatical relation between them, e.g., $(minister, det, the)$, $(the, det{-}of, minister)$, $(copied, subj, minister)$, $(minister, subj{-}of, copied)$.

The following would be a semantic space in style of Lin [1998] for our example sentence (3.0.1), if we restrict the model to the three basic relations `subject, predicate, object`.

$$\left(\begin{array}{cccc} & (subj, minister) & (pred, copied) & (obj, supervisor) \\ minister & 0 & 1 & 0 \\ copied & 1 & 0 & 1 \\ supervisor & 0 & 0 & 0 \end{array}\right) \tag{3.2.4}$$

Both Grefenstette and Lin pay attention to the direction of the graph and they differentiate, for example, between a subject and an object position of the same word. Thus, there could exist *minister* as a *subject* and *minister* as an *object* as context elements.

In more recent work, Padó and Lapata [2007] build on the above work and use a dependency-parsed corpus for the construction of their semantic space. They interpret a dependency parse as an *undirected* graph. A further difference of their model from the previous ones is that they allow a dependency path length of more than one, i.e. for longer constructions, and thereby integrate indirect semantic relations.

Padó and Lapata [2007] generalize their semantic space model by extending the formal model of Lowe [2001] to a quintuple with an additional element and three parameters $< T, B, M, S, A, cont, \mu, \upsilon >$ where:

T: - the extension - is the set of target words;

`T` can contain either word types or word tokens.

B: is the set of basis elements;

Basis elements are restricted by the allowed syntactic relations, e.g., only `subject, predicate and object` relations may be considered. Padó and Lapata [2007] explicitely define this function as a separate parameter (see below).

M: is the matrix $M = B \times T$;

A: is the lexical association function;

S: is the similarity measure;

The parameters are:

- $cont : T \rightarrow 2^{\Pi}$ is the content selection function;

  A content, or context, selection function maybe a function that considers only subject and object relations:

  $cont(t) = \pi \in \Pi_t | l(\pi) \in [V, subj, N], [V, obj, N]*$

- μ is the basis mapping function.

  Unlike the above mentioned models of Grefenstette [1994] and Lin [1998], the basis elements in Padó and Lapata [2007] can be mapped, e.g., to their *terminal* words, using the terminology of the graph theory. This kind of mapping would make the matrix look like word-based models (see the matrix (3.2.5)).

- v is the path value function.

  An example of the path value function would be a function that gives a numerical value of 1 to the path length of one and otherwise is the value inversely proportional to the path length.

$$\left(\begin{array}{cccc} & minister & copied & supervisor \\ minister & 0 & 1 & 0.5 \\ copied & 1 & 0 & 1 \\ supervisor & 0.5 & 1 & 0 \end{array}\right) \tag{3.2.5}$$

3.2.4 Distributional Memory Framework

Baroni and Lenci [2010] introduce a syntactically enriched model of a somewhat different nature than the above approaches - the Distributional Memory Framework.

Weighted triple structure T is the expected input to the model. A weighted triple structure is a ternary tuple or triple in the similar sense as Grefenstette [1994]; Lin [1998]; Padó and Lapata [2007] use it:

$$T \subseteq W_1 \times L \times W_2$$

These are tuples of two arguments W_1 and W_2 ordered by relations L that can be expressed in different ways. Two assumptions are made for the given moment: $W_1 = W_2$ and for any link l there exists an inverse link l^{-1} for the same two arguments.

Baroni and Lenci [2010] suggest three different types of models defined by different types of relations:

1. **DepDM**
2. **LexDM**
3. **TypeDM**

DepDM. Here, links are dependency relations that are obtained through dependency parse, similarly to the dependency - based semantic space models. Prepositions are represented by their lexical label.

For example: $< minister, SBJ\_INTR, copy >$, $< minister, VERB, supervisor >$, $< supervisor, FROM, copy >$.

Every triple also receives an inverse link: $< copy, SBJ\_INTR^{-1}, minister >$, $< supervisor, VERB^{-1}, minister >$, $< copy, FROM_{-1}, supervisor >$.

Local Mutual Information (LMI)[1] is used as a weighting function. All negative values are turned into 0. The resulting *DepDM* tensor has $30693 \times 796 \times 30693$ dimensions and density[2] of 0.0149%.

[1] $LMI = O_{ijk} log \frac{O_{ijk}}{E_{ijk}}$ where O_{ijk} is the co-occurrence count of the triple and E_{ijk} is the expected count under independence.

[2] the proportion of non-zero entries

LexDM is motivated by lexico-syntactic patterns in the work of Hearst [1992]. Relations are expressed here by complex links combining the dependency relation as in `DepDM` or, by frequent words, their lexical forms with suffixes encoding part of speech and morphological form in the following manner: $< minister, SBJ\_INTR + n - the, copy >$.

Similarly to `DepDM`, `LexDM` contains also inverse links and the weighting function is LMI. *LexDM* has a dimensionality of 30693 × 3352148 × 30693 with density 0.00001%.

TypeDM. The third model is `TypeDM`, which has a size of 30693 × 25336 × 30693 and a density of 0.0005%. Links here are defined by types of realizations not by their surface forms. Thus, the links are adopted here from the patterns of *LexDM* and the suffixes of the patterns are used to count the surface forms of the links: e.g., *copied from the minister*, *copied from a minister*, *copied from ministers*, *copied from ADJ ministers* and so on will turn into $< copy, FROM, minister >$ with a count of 4 in this case.

This model is at closest to the one suggested in this thesis, except that we do not assume any preprocessing and restriction to only certain kinds of links[1].

Similarly to the model suggested in this thesis, the triples are saved in a third-order tensor. The crucial difference, however, is that the tensor is used only as a placeholder for the semantic space model; while we use the tensor in the first line for further semantic processing.

All the semantic operations on the distributional memory in Baroni and Lenci [2010] are still performed with matrices. For that, different semantic spaces are constructed from the tensor by means of labeled matricization. Matricization rearranges a third-order tensor into a matrix (Dunlavy et al. [2011]). Thereby four semantic spaces can be gained:

1. word by link-word ($W1 \times LW2$)
2. word-word by link ($W1W2 \times L$)
3. word-link by word ($W1L \times W2$)
4. link by word-word ($L \times W1W2$)

[1] Both models, the one of Baroni and Lenci [2010] and Giesbrecht [2010] have been suggested at the same time.

All further operations on matrices and semantic tasks are accomplished here in the similar way as in matrix-based approaches.

3.2.5 Tensor Approaches

Early attempts to apply higher-order tensors instead of vectors to text data came from research in information retrieval. Among them is the work of Liu et al. [2005] who show that a tensor space model is consistently better than a vector space model for text classification. Cai et al. [2006] suggest a 3-dimensional representation for documents and evaluate the model on the task of document clustering on Reuters-21578 corpus[1].

The above as well as a couple of further activities in this area in information retrieval research are less interested in the question of an adequate conversion of natural language text into the tensor. Most of them still use a vector-based representation as the basis and then mathematically convert vectors into tensors, without linguistic justification of such a transformation; or they use metadata as a third dimension. For example, Sun et al. [2006] employ an AUTHOR × KEYWORD × DATE tensor, or Chew et al. [2007] use a tensor-based model for cross-language information retrieval, with language as the third dimension. Franz et al. [2009] model semantic RDF graphs by a 3-dimensional tensor that enables the seamless representation of arbitrary semantic links for authority ranking in Semantic Web applications.

However, the use of matrix- and tensor-based representation for modeling meaning does not necessarily count as a semantic space model. The main defining property of the latter as defined in this work is that the values of matrix entries are directly determined from word distribution patterns in text without using external metadata.

Turney [2007] is one of the few to study the application of tensors to semantic space models. However, the emphasis of that work was more on the evaluation of different tensor decomposition methods for such spaces than on the model of text representation in three dimensions per se. However, he suggests in this paper a three-dimensional tensor having words and their connecting patterns as dimensions[2].

Van de Cruys [2009, 2010] uses a dependency parsed Dutch corpus and builds a three dimensional tensor consisting of `subjects`, `verbs` and `direct objects` for the concrete task of determining words' selectional preferences. This work is indeed at

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

[2] `LexDM` in Baroni and Lenci [2010] is similar to Turney's model of 2007.

closest in the suggested approach to ours; even though the original motivation is quite different and the ideas emerged in parallel.

3.2.6 Summary

Lots of different semantic space models have been suggested since the 1990s. In this chapter, we've offered an overview of the most influential models. A semantic space model per se has been formalized by Lowe [2001] as a quadruple, consisting of the basis elements that define the context ($< B >$), a mapping between basis and target elements ($< A >$), a similarity measure ($< S >$) and a possible transformation of the original co-occurrence matrix ($< M >$).

The components $< A, S, M >$ are used interchangeably in the meanwhile and have been tested in all possible combinations with lots of forms of semantic space models. A deeper analysis of the influence of weighting schemes as well as the choice of similarity measure, the chosen transformation, the number of dimensions and other factors that can influence the performance of such models is out of the scope of this thesis and there is a lot of literature offering these insights [e.g. Nakov et al., 2001, 2003].

All in one, word-based and syntactically-enriched models differ mostly in the way in which the original matrix is built ($< B >$) which is either a WSM- or a HAL-way, i.e. in one direction or in both ways, with many possible context window sizes and with possible use of filtering in the form of stop words or syntax.

In 2009 and 2010, three-way tensor-based approaches started gaining in importance in computational linguistics.

Baroni and Lenci [2010] suggested to use a tensor as a placeholder and unifying framework for distributional models, so that there is a "one-fits-all" model that can be mathematically transformed into different kinds of matrices depending on the task. Hence, there is no need any more to build separate distributional models for different tasks, as it is traditionally done in this kind of research.

The work of Van de Cruys [2009], applying a tensor model, is indeed inspired by a concrete task - selectional preferences - and the tensor model itself is not further elaborated except for the task in question.

In both cases, the model is created from a syntactical or pattern-based preprocessing. Furthermore, only Van de Cruys [2009] uses tensor-based operations for semantic processing.

The model suggested in this thesis and defined in Chapter 4 does not need any explicit preprocessing except for the purpose of reducing the computational demands. It is based completely on the traditions of early distributional semantics. Nevertheless, it offers a linguistically adequate framework for representing semantics by integrating word order information, like HAL-based models, and at the same time allowing for including more structural information into the model by representing words by means of matrices instead of vectors. Furthermore, having represented words as matrices, we can apply a matrix multiplication operator to compose words into phrases or sentences.

The definition of the problem of compositionality in vector spaces as well as a review of existing approaches is following in the next section.

3.3 Compositionality in Semantic Space Models

The principle of compositionality, commonly atributed to Frege [1884] and first formalized by Montague [1974], claims that the meaning of a phrase or a sentence is determined by its structure and the meaning of its components where the meaning is interpreted as the notion of truth [Szabó, 2012].

Compositionality has traditionally been an issue in formal approaches to natural language. Symbolic or logical approaches to meaning, following the traditions of Montague's semantics, have been concerned prevalently with functional composition of individual units into sentences and not with the meaning of those individual units.

The semantic space models, in contrary, were predominantly used for meaning representation of single words and thereby have been the mainstream of interest in lexical semantics since the 90s. Until recently, little attention has been paid to the way of modelling more complex conceptual structures with such models, which is a crucial barrier for semantic vector models on the way to model language [Widdows, 2008]). As a consequence, an emerging area of research that receives more and more attention among the advocates of distributional models are the methods, algorithms and evaluation strategies for modeling of compositional meaning within the framework of distributional semantics.

To summarize, according to the principle of compositionality the meaning of a phrase or a sentence is defined by:

1. the meaning of component words that are represented as distributional vectors in semantic space models and
2. the way they are combined.

Word representation. In respect to word meaning modeling, there are two general trends (cf. Baroni et al. [2013]): constructing "word meaning in context" [see Erk and Padó, 2008] versus picking out the right word meaning in the process of composition [see Mitchell and Lapata, 2008].

The works of Erk and Padó [2008]; Thater et al. [2010, 2011] belong to the first group of approaches. Dinu, Thater, and Laue [2012] offer an overview of these methods and prove that they are conceptually equivalent in that they component-wise multiply the second order vector of one word (be it a target or a context word) with the first order vector of another word.

Such approaches use composition as a kind of auxiliary means to restrict word's meaning. This topic per se is out if the scope of our current work and can be seen as complementary.

There are two ways to address word meaning representation within both of these trends. Words can be represented as **types** or **lemmas** summing up all the occurrences in one representation [cf. Mitchell and Lapata, 2008]. This kind of approaches is often called **type-based**. Another way would be to differentiate between single concrete uses of words. This kind of methodology is usually defined as **token-based**.

Works of Schütze [1998]; Katz and Giesbrecht [2006]; Erk and Padó [2010]; Reisinger and Mooney [2010]; Reddy, Klapaftis, McCarthy, and Manandhar [2011] provide examples of **token-based approaches**.

Schütze [1998]; Reisinger and Mooney [2010]; Reddy, Klapaftis, McCarthy, and Manandhar [2011] cluster words' contexts to produce groups of similar contexts in order to build a "prototype" vector of each cluster which is its centroid.

Erk and Padó [2010]; Reddy et al. [2011] furthermore give up the one-vector-per-word paradigm and turn to an "exemplar-based model" motivated by cognitive psychology research. Here a word is represented by sets of similar context examples instead of one prototype that is "representative" for this set. Reisinger and Mooney [2010] call those **multi - prototype vectors**.

Similarly to our initial idea, it has been recognized by the above approaches that a single vector is not enough to represent word meaning, some of them realized it by "exemplar-based approaches" within a token-based word meaning representation paradigm [Erk and Padó, 2010; Reddy, Klapaftis, McCarthy, and Manandhar, 2011]; the others, such as Erk and Padó [2008]; Thater, Fürstenau, and Pinkal [2010] by adding vectors for the words selectional preferences to represent the word type.

We argue, similarly to Erk and Padó [2008], that single vectors are too weak to represent the word meaning as a vector can encode only a limited and fixed amount of structural information, and it is difficult to foresee how deeper semantic or syntactic structures, like predicate-argument, can be encoded into a vector.

To summarize, the above arguments leave us with two open questions in respect to word meaning representation. The first one is whether there should be a single representation per word (type-based) or whether multiple representations are allowed (token-based). The second problem is whether a vector is good enough to ”model” word’s meaning.

Interestingly, the question of **word representation**, especially of polysemous words, **in the brain** is also still an open issue in cognitive research. Most meaningful words in natural language are polysemous to some degree, and according to Zipf’s law [Zipf, 1935] the more frequent words are, the more polysemous they tend to be [Pylkknen et al., 2006].

Consequently, a question arises: are *(wedding) rings* and *(boxing) rings* represented by one lexical entry in the mental lexicon, or how many *banks* are there? *Ring* would be an example of polysemy in natural language, i.e. when different meanings of a word are still related in some way, and *bank* would be a homonym, i.e. a word having several unrelated meanings, e.g., *bank as a financial institute* and *bank of the river.*

Pylkknen et al. [2006] show empirical results from a combination of behavioral and magnetoencephalographic (MEG) measurements on experiments with polysemous words and demonstrate that such words, i.e. words having multiple related meanings and identical lexical representations, form part of a single lexical entry. Concerning homonyms, the researchers favour more the opposite hypothesis, i.e., that they are represented by several entries in the mind. For example, Tamminen et al. [2006] show by means of the Psychological Refractory Period (PRP) logic in an auditory lexical decision study that ”the ambiguity between unrelated meanings is being resolved at an early stage, the ambiguity between related senses is resolved at a later stage”. Such findings motivated researchers (e.g., Rodd et al. [2004]) to build computational models of meaning where homonyms are represented by several representations for their different meanings.

Still both options are realistic and open in the end, so we can currently just assume that it is one way or another. Until then, both ways of exploration are perfectly valid.

Operator for composition. Concerning the second aspect of compositionality models, i.e. the way of combination, two tendencies have established themselves among the advocates of distributional semantics: the early efforts have concentrated on finding an optimal mathematical operation to reproduce compositionality within a purely distributional paradigm; the later trends are turning back to formal semantics and the ways to combine formal approaches and distributional semantics.

Until recently, the "bag-of-words" (BOW) approach has been used as a default to get the meaning of phrases and sentences in vector spaces (Landauer and Dumais [1997], Deerwester et al. [1990]). BOW consists of simply adding up the individual vectors of the words independent of their representation in the word space to get the meaning of a phrase or a sentence. Thus, with such approaches two sentences - *"The student copied from his supervisor"* and *"The supervisor copied from his student"* - would mean the same. Vector summation operations can not serve as an adequate means of semantic composition, as word order information is ignored and the meaning of the whole is an average of its parts. Presumably, one of the few statements on which most researchers would indeed agree is that compositional meaning is not purely an average of its component meanings.

Cruse [2000] postulates two main modes of meaning composition: additive and interactive. An example of additive combination would be in a sentence like: *"**A teacher and five students** were in the classroom"*. An interactive mode of combination implies that one of the components' semantics is changed through composition. Here two variants are possible: the resulting meaning is similar to one of the components (*a big boy* is a kind of *boy*), or it becomes completely unrelated to any of the parts, as in idioms for example.

Hopefully, in the traditions of cognitive science, computational models of meaning and compositionality will help to understand the ways, human brain is performing the operation of word meaning combination.

Within every of the mentioned trends, several computational approaches have been suggested that will be briefly reviewed in the following sections.

3.3.1 Mathematical Compositionality Operators

Since the problem of modeling compositionality has become more announced in distributional semantics, the researchers have come up with a number of mathematical models for linguistic composition. Those can be classified into four main groups [Giesbrecht, 2009].

Let *w1w2* denote the composition of two vectors *w1* and *w2*. The estimated compositional meaning vector *w1w2* is calculated by taking it to be:

1. the sum of the meaning vectors of the parts, i.e., the compositional meaning of an expression *w1w2* consisting of two words is taken to be sum of the meaning vectors for the constituent words *w1* and *w2*: $(w1w2)_i = w1_i + w2_i$;

   Thus, the "compositional" vector for *yellow press* in this case would be the sum of the vectors for *yellow* and *press*.

2. the simplified multiplicative model as it is defined in Mitchell and Lapata [2008]: under the assumption that only the ith component of *w1* and *w2* contribute to the ith component of *w1w2*, we can formulate vector multiplication operation as: $(w1w2)_i = w1_i \cdot w2_i$;

   The multiplication model seems to be more linguistically adequate by "allowing the content of one vector to pick out the relevant content of the other" [Mitchell and Lapata, 2008].

3. the tensor product: if the vector of the word *w1* has components $w1_i$ and the vector of the word *w2* has components $w2_j$, then the tensor product $(w1 \otimes w2)$ is a matrix whose ij^{th} entry is $w1_i w2_j$ (cf. Widdows [2008]);

   The first usage of a tensor product as a means of vector composition is usually attributed to Smolensky [1990]. Many researchers see the problem in the dimensionality of the resulting product of two vectors which is a matrix. That is why circular convolution has been proposed.

4. the convolution product, which is also a kind of vector multiplication that results in the third vector of dimensionality $(m + n - 1)$. Given two vectors $w1 = [w1_1, w1_2, w1_{...}, w1_m]$ and $w2 = [w2_1, w2_2, w2_{...}, w2_n]$, their convolution $(w1 * w2)$ is defined as $(w1w2)_i = \sum_j w1_j w2_{i-j+1}$.

   Circular convolution (Holographic Reduced Representations) was first suggested by Plate [1991, 1995] as a means of compositional distributed representation.

The most cited work for this class of approaches, which deserves to be mentioned separately here, is the work of Mitchell and Lapata [2008], further elaborated in Mitchell and Lapata [2010], who present a framework for compositionality representation with distributional vector space model. They propose several variations of multiplicative and additive models that are generalized in the following formula:

$$p = f(u, v, R, K)$$

- where p stands for a composed vector from u and v connected by some syntactic relation R, and K represents any available additional knowledge which may be relevant for the process for composition.

If we fix R to a certain relation, e.g., *an adjective-noun phrase*, and ignore aditional knowledge K that may or may not be available, the above formula is reduced to a binary function $p = f(u, v)$. Assuming that f is a linear function they end up with two major classes of composition:

additive models : $p = Au + Bv$ where A and B are matrices that define the contribution of u and v to p.

multiplicative models : $p = Cuv$ where C is a tensor of rank 3 that projects a tensor product of u and v into p.

Both A, B and C components allow to integrate the influence of syntax into the model. If those factors are ignored and symmetry is allowed, we end up with simple addition and multiplication models which are both commutative.

In order to avoid the drawback of components with zero values in multiplicative models, they also suggest to combine a multiplicative and additive model, e.g., in the following way: $p_i = \alpha u_i + \beta v_i + \gamma u_i v_i$ where α, β and γ are weighting constants.

3.3.2 Integrating Formal and Distributional Semantics

A further class of approaches to modeling compositionality is motivated by *formal semantics* [Montague, 1974].

Clark and Pulman [2007] suggest to combine symbolic representation of a sentence (e.g. a parse tree or a dependency graph) with distributional word vectors by means of a tensor product. For example, such a representation for a sentence *A boy kicked the ball* could look like this:

$$kick \otimes subj \otimes boy \otimes obj \otimes ball$$

- where $\otimes$ stands for tensor product. The question of obtaining vectors for dependency relations, like *subject* or *object*, is left open here.

They extend and formalize this framework in Clark et al. [2008] using Lambek's pregroup semantics [Lambek, 1999].

Coecke et al. [2010] proposes a mathematical model for composition that unifies distributional vectors and a compositional theory of grammatical types based on category theory, and the compositional meaning of a sentence is also modeled as a function of tensor products on distributional vectors.

Clarke [2012] suggests a context-theoretic framework aiming at a combination of formal semantics and vector-space models of word representation. This work was intended as a purely theoretical framework that is restrictive in respect to the set of possibilities a theory of meaning need to hold.

Functional Approaches

In *formal semantics*, the meaning of certain classes of words is modelled as a function, e.g., the meaning of attributive adjectives, like *large table* can be modelled as an intersection of adjectives and nouns:

$$[large\ table] = \{large\_objects\} \bigcap \{tables\}$$

Combining this view of composition as in *formal semantics* and the way to represent word meaning as in distributional semantics, the compositional distributional meaning of an *adjective-noun pair "large table"* can be learnt from the corpus of text data by considering distributional noun and adjective vectors for *table* and *large* respectively as well as the vector for the phrase *large table* as a whole [Guevara, 2010].

Such approaches use machine learning, in particular regression analysis, to model compositionality. Thus, Guevara [2010]; Baroni and Zamparelli [2010] use regression analysis to model the compositionality of *adjective-noun (AN)* constructions. Later, Guevara [2011] extends his approach to further *verb-noun* combinations. Baroni et al. [2013] model nouns, determiner phrases and sentences as vectors, whereas, for example, adjectives, verbs, determiners, prepositions are modeled as functions on those vectors.

Regression analysis is a way to model the influence of one or more independent variables onto a dependent variable. In case of natural language semantics, words are *independent variables* and phrases or sentences are *dependent variables*.

Thus, the compositional meaning of a phrase can be computed by means of multivariate multiple linear regression: $Av1+Bv2 = v3$ (Guevara [2010, 2011]), where $v1$ and $v2$ are input distributional vectors of component words, and A and B are weight matrices that are supposed to map the contribution of input components to the resulting phrase. Regression is used to estimate the weight matrices.

In Baroni and Zamparelli [2010], nouns are represented as distributional vectors and adjectives are linear functions encoded as matrices. Like Guevara [2010], they estimate the values in the weight matrix by partial least squares regression. The difference is that Baroni and Zamparelli [2010] use only distributional vectors of the component nouns as input for training the model, or in terms of regression analysis as independent variables, whereas the target phrase vectors are outputs, or dependent variables. Furthermore, in contrast to Guevara [2010], separate models are trained for each adjective and for the same adjective in different grammatical positions, e.g., in attributive or predicative position.

Grefenstette and Sadrzadeh [2011a]; Grefenstette et al. [2011]; Grefenstette and Sadrzadeh [2011b] implement the theory suggested in Coecke et al. [2010] and also state that not everything can be a vector in a semantic space; some objects, like *verbs*, are functions that can be modeled by tensors. Here, as in all formal semantics driven approaches, syntax determines the composition. Tensors, i.e. functions, are learnt from the corpus, similarly to Guevara [2010] and Baroni and Zamparelli [2010].

Grefenstette and Sadrzadeh [2011b] demonstrate this functional approach on intransitive and transitive verbs, while Baroni and Zamparelli [2010]; Guevara [2010] for adjective-noun constructions.

Baroni, Bernardi, and Zamparelli [2013] extend the functional approach to the representation of simple intransitive ("The fire glowed") and transitive senteces ("Table shows results") and generalize their framework of compositional distributional semantics by means of *multi-step regression learning* for tensors of rank 3 and more.

By means of example from Baroni, Bernardi, and Zamparelli [2013]: in order to estimate a tensor for *eat* first the matrices for *eat meat* and *eat pie* are learnt by regression from corpus examples (*"Dogs eat meat. Cats eat meat."*) and a matrix for *eat pie* from examples like *"Boys eat pie. Girls eat pie."*. The tensor for *eat* is then estimated by regression from the vectors for *meat* and *pie* as input and the matrices for *eat meat* and *eat pie* as output.

Socher et al. [2012] propose, unlike the above mentioned work, to use non-linear functions and present a novel recursive neural network model for computing semantic compositionality. Within this framework every word is represented by a vector and a matrix, where a vector contains the meaning of a word and the matrix reproduces how a word modifies the meaning of the other words it combines with.

Turney [2012] suggests a dual-space model that consists of a space for measuring domain similarity and another one for measuring function similarity.

3.4 Conclusion and Outlook

Chapter 3 offers a brief overview of existing semantic space models as well as presents approaches to the problem of compositionality within the distributional semantics paradigm.

Many of the above discussed works claim to present general frameworks for either distributional semantics or for compositionality.

However, most of the available models in distributional semantics have been optimized or constructed specifically for one of the above tasks, except for the work of Baroni and Lenci [2010]. It is worth mentioning, that the latter was published at the same time as our work on matrices [Giesbrecht, 2010; Rudolph and Giesbrecht, 2010].

As the review of the related literature shows, a number of models has been suggested to solve either the problem of word order integration or the task of compositionality. The models of compositionality including advanced linguistic preprocessing, such as dependency parsing, automatically solve the problem of word order, but require good performance of computational methods for these preprocessing steps. The state-of-the-art[1] accuracy for automatic dependency parsing is below 90% for English and even worse for other languages, if available at all[2].

Furthermore, these methods need many training instances per phrase to get reliable results, and usually separate training is required for every kind of expression or combination of those. Due to the productivity[3] of natural language, it is hardly imaginable to foresee that such training examples can be always available.

[1] 31.December 2013

[2] The model that is developed in this thesis is language-independent.

[3] Productivity is the ability to create unlimited number of word combinations and sentences that may have been never heard before.

We argue that a matrix-based representation allows us to integrate contextual information as well as model compositionality in a more general manner.

Our goal is not to get an optimal performance on a certain task. We are more after a linguistically and mathematically adequate model of semantics that reflects the insights from cognitive research and that is equally suited for most of the semantic processing tasks. We perfectly realize, that this is an ambitious task and we do not aim at claiming that our current model can do it all; but it offers a *promising* venue.

4

Distributional Tensor Space Model (DTSM)

In the following we describe the Distributional Tensor Space Model as well as theoretical foundations of Compositional Matrix Space Model.

We first formulate DTSM in terms of the formalism presented by Lowe [2001]. Furthermore, we extend this formalism from a quadruple to a sextuple: $\langle A, B, S, M, T, C\rangle$ where C is a compositionality operator and T is the representation of target words[1]. Traditionally it has been assumed that T is just a vector. Since recently it has been recognized that one vector is not enough.

4.1 Three-Way Model of Distributional Semantics

Motivated by the ideas of distributional semantics and the mathematics behind it, we propose a novel approach that offers a potential of both integrating syntax and a mathematically and linguistically justified composition operation into vector space models by employing a 3-dimensional model that accounts for order-dependent word contexts and assigns to words characteristic matrices such that semantic composition can be later realized in a natural way via matrix multiplication.

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-

[1] Padó and Lapata [2007] also extended Lowe's definition by adding T (target words) and a number of further parameters. However, for them T is just a set of target words. See Chapter 3.2.3 for more details.

occurrences on different axes, in contrast to 2-dimensional models, in order 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”. Assuming a context window $w = 3$ and the prior stop words removal, we obtain a $5 \times 5 \times 5$ tensor. It would be hardly comprehensible if we visualize all three axes with all the contexts on a piece of paper, so we reproduce two middle (Y) slices of the resulting tensor in Tables 4.1 and 4.2.

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

Table 4.1: Slice for target word KICK

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 4.2: Slice for target word BALL

The first table shows the middle matrix for the word *KICK* and the second table is the matrix for the word *BALL*. The words on the left (rows) display the left contexts of *KICK* or *BALL*; those on the right - the right contexts correspondingly. The interpretation is straightforward: everywhere, where there is an entry other than zero in the table, there exists a triple co-occurrence in the text - $\langle Peter, kick, ball\rangle$, $\langle Paul, kick, ball\rangle$,

$\langle Paul, kick, Peter\rangle$ in the *KICK*-Table occur just once, and $\langle kick, ball, slowly\rangle$ (*BALL*-Table) occurs twice in the text.

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.

Similarly to traditional vector-based distributional models, dimensionality reduction needs to be performed in three dimensions either, as the resulting tensor is even sparser than its two-way analogues (see the examples of *KICK* and *BALL*). To this end, we employ an analogue of singular value decomposition for three dimensions, as introduced in Section 2.

It is irrelevant for the moment which of the three tensor decomposition methods we employ. We leave a more thorough exploration of the effects of concrete dimensionality reduction algorithms as well as finding out a mathematically sound number of dimensions to future research. In this work we follow the traditions of empirical research and fix an optimal number of dimensions by means of the algorithms performance on concrete tasks. We test all three methods defined in Chapter 2 on a small portion of a corpus and make a decision in the end in favour of **non-negative tensor factorization (NTF)** as a major decomposition method for this work.

Following the formalism suggested by Lowe [2001], we can define our models as a sextuple $\langle A, B, S, M, T, C\rangle$:

B is defined by words to the left and to the right within a predefined context window and constrained by sentence boundaries, of which only a certain number (e.g., 2000 or 5000) are filtered;

Filtering may be realized either by most frequent words in the corpus, or by middle frequency words, or by using only certain parts of speech, or any other heuristics.

For example, **B** in Tables 4.1 and 4.2 would be *Peter, Paul, kick, ball, slowly.*

A is a weighting function from target to basis words;

After evaluating three weighting functions, such as *frequency of occurrence*, *boolean* and *pointwise mutual information (PMI)*, we choose to use the later as the best performing one.

PMI is a measure of association between (usually) two variables in statistics or two words in linguistics. In our case, we need to generalize PMI as follows for triples:

$$pmi(x, y, z) = log \frac{O_{x,y,z}}{E_{x,y,z}}$$

where $O_{x,y,z}$ is the number of co-occurrence of words x, y, z in the corpus and $E_{x,y,z}$ is the expected frequency.

In our case, $E_{x,y,z}$ is computed by multiplying the occurrences of the left word in the triple in all left contexts in the corpus, the middle word in the triple in all middle contexts in the corpus, and the right word in the triple in all right contexts in the corpus.

In our reference Tables 4.1 and 4.2, **A** is simple frequency, that is, how often the words co-occur.

S is determined by a cosine similarity measure between matrices as defined in Chapter 2;

M is represented by tensor decomposition; in our case it is non-negative tensor factorization (NTF);

C is matrix multiplication (cf. Section 4.2) and, additionally, matrix addition;

T is a matrix consisting of the left and right word co-occurrences.

For example, two matrices for target words *KICK* and *BALL* are presented in Tables 4.1 and 4.2

Hence, we extend the original quadruple to the sextuple; with two additional elements C (compositionality framework) and T (the representation of target words).

4.2 Compositional Matrix Space Model (CMSM) as a Framework for Language Compositionality

The underlying principle of compositional semantics is that the meaning of a sentence (or a word phrase) can be derived from the meaning of its constituent tokens by applying a composition operation. More formally, the underlying idea can be described as follows:

given a mapping $[\![\, \cdot \,]\!] : \Sigma \to \mathbb{S}$ from a set of tokens (words) Σ into some semantical space $\mathbb{S}$ (the elements of which we will simply call "meanings"), we find a semantic composition operation $\bowtie: \mathbb{S}^* \to \mathbb{S}$ mapping sequences of meanings to meanings such that the meaning of a sequence of tokens $\sigma_1\sigma_2 \dots \sigma_n$ can be obtained by applying $\bowtie$ to the sequence $[\![\sigma_1]\!][\![\sigma_2]\!] \dots [\![\sigma_n]\!]$. This situation qualifies $[\![\cdot]\!]$ as a homomorphism between $(\Sigma^*, \cdot)$ and $(\mathbb{S}, \bowtie)$ and can be displayed as follows:

$$\begin{tikzcd}
\sigma_1 \arrow[rrrr, bend left, "\text{concatenation } \cdot"] \arrow[d, "{[\![\cdot]\!]}"] & \sigma_2 \arrow[rrr, bend left] \arrow[d, "{[\![\cdot]\!]}"] & \cdots & \sigma_n \arrow[r, bend left] \arrow[d, "{[\![\cdot]\!]}"] & \sigma_1\sigma_2 \dots \sigma_n \arrow[d, "{[\![\cdot]\!]}"] \\
{[\![\sigma_1]\!]} \arrow[rrrr, bend right, "\text{composition } \bowtie"'] & {[\![\sigma_2]\!]} \arrow[rrr, bend right] & \cdots & {[\![\sigma_n]\!]} \arrow[r, bend right] & {[\![\sigma_1\sigma_2 \dots \sigma_n]\!]}
\end{tikzcd}$$

A great variety of linguistic models are subsumed by this general idea ranging from purely symbolic approaches (like type systems and categorial grammars) to rather statistical models (like vector space and word space models). At the first glance, the underlying encodings of word semantics as well as the composition operations differ significantly. However, we argue that a great variety of them can be incorporated – and even freely inter-combined – into a unified model where the semantics of simple tokens and complex phrases is expressed by matrices and the composition operation is standard matrix multiplication.

More precisely, in Compositional Marix-Space Models (CMSM) [Rudolph and Giesbrecht, 2010], we have $\mathbb{S} = \mathbb{R}^{n \times n}$, i.e. the semantical space consists of quadratic matrices, and the composition operator $\bowtie$ coincides with matrix multiplication as introduced in Chapter 2.

In the following, we will provide diverse arguments illustrating that CMSMs are intuitive and natural.

4.2.1 Algebraic Plausibility – Structural Operation Properties

Most linear-algebra-based operations that have been proposed to model composition in language models are associative and commutative. Thereby, they realize a multiset (or bag-of-words) semantics that makes them insensitive to structural differences of phrases conveyed through word order.

While associativity seems somewhat acceptable and could be defended by pointing to the stream-like, sequential nature of language, commutativity seems way less justifiable, arguably.

As mentioned before, matrix multiplication is associative but non-commutative, hence we propose it as more adequate for modeling compositional semantics of language.

4.2.2 Neurological Plausibility – Progression of Mental States

From a very abstract and simplified perspective, CMSMs can also be justified neurologically.

Suppose the mental state of a person at one specific moment in time can be encoded by a vector $\mathbf{v}$ of numerical values; one might, e.g., think of the level of excitation of neurons. Then, an external stimulus or signal, such as a perceived word, will result in a change of the mental state. Thus, the external stimulus can be seen as a function being applied to $\mathbf{v}$ yielding as result the vector $\mathbf{v}'$ that corresponds to the person's mental state after receiving the signal. Therefore, it seems sensible to associate with every signal (in our case: token σ) a respective function (a linear mapping, represented by a matrix $M = [\![\sigma]\!]$ that maps mental states to mental states (i.e. vectors $\mathbf{v}$ to vectors $\mathbf{v}' = \mathbf{v}M$).

Consequently, the subsequent reception of inputs σ, σ' associated to matrices M and M' will transform a mental vector $\mathbf{v}$ into the vector $(\mathbf{v}M)M'$ which by associativity equals $\mathbf{v}(MM')$. Therefore, MM' represents the mental state transition triggered by the signal sequence $\sigma\sigma'$. Naturally, this consideration carries over to sequences of arbitrary length. This way, abstracting from specific initial mental state vectors, our semantic space $\mathbb{S}$ can be seen as a function space of mental transformations represented by matrices, whereby matrix multiplication realizes subsequent execution of those transformations triggered by the input token sequence.

4.2.3 Psychological Plausibility – Operations on Working Memory

A structurally very similar argument can be provided on another cognitive explanatory level. There have been extensive studies about human language processing justifying the hypothesis of a *working memory* [Baddeley, 2003]. The mental state vector can be seen as representation of a person's working memory which gets transformed by external input. Note that matrices can perform standard memory operations such as storing, deleting, copying etc. For instance, the matrix $M_{\text{copy(k,l)}}$ defined by

$$M_{\text{copy(k,l)}}(i,j) = \begin{cases} 1 \text{ if } i = j \neq l \text{ or } i = k,\, j = l, \\ 0 \text{ otherwise.} \end{cases}$$

applied to a vector $\mathbf{v}$, will copy its kth entry to the lth position. This mechanism of storage and insertion can, e.g., be used to simulate simple forms of anaphora resolution.

4.2.4 CMSMs Encode Vector Space Models

In VSMs numerous vector operations have been used to model composition [Widdows, 2008], some of the more advanced ones being related to quantum mechanics. We show how these common composition operators can be modeled by CMSMs.[1] Given a vector composition operation $\bowtie$: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, we provide a surjective function $\psi_{\bowtie} : \mathbb{R}^n \to \mathbb{R}^{n' \times n'}$ that translates the vector representation into a matrix representation in a way such that for all $\mathbf{v}_1, \ldots \mathbf{v}_k \in \mathbb{R}^n$ holds

$$\mathbf{v}_1 \bowtie \ldots \bowtie \mathbf{v}_k = \psi_{\bowtie}^{-1}(\psi_{\bowtie}(\mathbf{v}_1) \ldots \psi_{\bowtie}(\mathbf{v}_k))$$

where $\psi_{\bowtie}(\mathbf{v}_i)\psi_{\bowtie}(\mathbf{v}_j)$ denotes matrix multiplication of the matrices assigned to $\mathbf{v}_i$ and $\mathbf{v}_j$.

4.2.4.1 Vector Addition

As a simple basic model for semantic composition, vector addition has been proposed. Thereby, tokens σ get assigned (usually high-dimensional) vectors $\mathbf{v}_\sigma$ and to obtain a representation of the meaning of a phrase or a sentence $w = \sigma_1 \ldots \sigma_k$, the vector sum of the vectors associated to the constituent tokens is calculated: $\mathbf{v}_w = \sum_{i=1}^{k} \mathbf{v}_{\sigma_i}$.

This kind of composition operation is subsumed by CMSMs; suppose in the original model, a token σ gets assigned the vector $\mathbf{v}_\sigma$, then by defining

$$\psi_+(\mathbf{v}_\sigma) = \left(\begin{array}{ccc|c} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 1 & 0 \\ \hline \multicolumn{3}{c|}{\mathbf{v}_\sigma} & 1 \end{array}\right)$$

[1] In our investigations we will focus on VSM composition operations which preserve the format (i.e. which yield a vector of the same dimensionality), as our notion of compositionality requires models that allow for iterated composition. In particular, this rules out dot product and tensor product. However the convolution product can be seen as a condensed version of the tensor product.

(mapping n-dimensional vectors to $(n+1) \times (n+1)$ matrices), we obtain for a phrase $w = \sigma_1 \dots \sigma_k$

$$\psi_+^{-1}(\psi_+(\mathbf{v}_{\sigma_1}) \dots \psi_+(\mathbf{v}_{\sigma_k})) = \mathbf{v}_{\sigma_1} + \dots + \mathbf{v}_{\sigma_k} = \mathbf{v}_w.$$

Proof. By induction on k. For $k = 1$, we have $\mathbf{v}_w = \mathbf{v}_\sigma = \psi_+^{-1}(\psi_+(\mathbf{v}_{\sigma_1}))$. For $k > 1$, we have

$$\begin{aligned}
& \psi_+^{-1}(\psi_+(\mathbf{v}_{\sigma_1}) \dots \psi_+(\mathbf{v}_{\sigma_k - 1}) \psi_+(\mathbf{v}_{\sigma_k})) \\
= \quad & \psi_+^{-1}(\psi_+(\psi_+^{-1}(\psi_+(\mathbf{v}_{\sigma_1}) \dots \psi_+(\mathbf{v}_{\sigma_k - 1})))\psi_+(\mathbf{v}_{\sigma_k})) \\
\overset{i.h.}{=} \quad & \psi_+^{-1}(\psi_+(\textstyle\sum_{i=1}^{k-1} \mathbf{v}_{\sigma_i})\psi_+(\mathbf{v}_{\sigma_k}))
\end{aligned}$$

$$= \psi_+^{-1}\left(\left(\begin{array}{ccc|c} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 1 & 0 \\ \hline \sum_{i=1}^{k-1}\mathbf{v}_{\sigma_i}(1) & \cdots & \sum_{i=1}^{k-1}\mathbf{v}_{\sigma_i}(n) & 1 \end{array}\right)\left(\begin{array}{ccc|c} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 1 & 0 \\ \hline \mathbf{v}_{\sigma_k}(1) & \cdots & \mathbf{v}_{\sigma_k}(n) & 1 \end{array}\right)\right)$$

$$= \psi_+^{-1}\left(\begin{array}{ccc|c} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 1 & 0 \\ \hline \sum_{i=1}^{k}\mathbf{v}_{\sigma_i}(1) & \cdots & \sum_{i=1}^{k}\mathbf{v}_{\sigma_i}(n) & 1 \end{array}\right) = \sum_{i=1}^{k} \mathbf{v}_{\sigma_i}$$

q.e.d.[1]

4.2.4.2 Component-wise Multiplication

On the other hand, the Hadamard product (also called entry-wise product, denoted by $\odot$) has been proposed as an alternative way of semantically composing token vectors.

By using a different encoding into matrices, CMSMs can simulate this type of composition operation as well. By letting

$$\psi_\odot(\mathbf{v}_\sigma) = \begin{pmatrix} \mathbf{v}_\sigma(1) & 0 & \cdots & 0 \\ 0 & \mathbf{v}_\sigma(2) & & \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{v}_\sigma(n) \end{pmatrix},$$

we obtain an $n \times n$ matrix representation for which $\psi_\odot^{-1}(\psi_\odot(\mathbf{v}_{\sigma_1}) \dots \psi_\odot(\mathbf{v}_{\sigma_k})) = \mathbf{v}_{\sigma_1} \odot \dots \odot \mathbf{v}_{\sigma_k} = \mathbf{v}_w$.

4.2.4.3 Holographic Reduced Representations

Holographic reduced representations as introduced by Plate [1995] can be seen as a refinement of convolution products with the benefit of preserving dimensionality: given

[1] The proofs for the respective correspondences for $\odot$ and $\circledast$ as well as the permutation-based approach in the following sections are structurally analog, hence, we will omit them for space reasons.

two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^n$, their *circular convolution product* $\mathbf{v}_1 \circledast \mathbf{v}_2$ is again an n-dimensional vector $\mathbf{v}_3$ defined by

$$\mathbf{v}_3(i+1) = \sum_{k=0}^{n-1} \mathbf{v}_1(k+1) \cdot \mathbf{v}_2((i-k \mod n)+1)$$

for $0 \leq i \leq n-1$. Now let $\psi_{\circledast}(\mathbf{v})$ be the $n \times n$ matrix M with

$$M(i,j) = \mathbf{v}((j-i \mod n)+1).$$

In the 3-dimensional case, this would result in

$$\psi_{\circledast}(\mathbf{v}(1) \quad \mathbf{v}(2) \quad \mathbf{v}(3)) = \begin{pmatrix} \mathbf{v}(1) & \mathbf{v}(2) & \mathbf{v}(3) \\ \mathbf{v}(3) & \mathbf{v}(1) & \mathbf{v}(2) \\ \mathbf{v}(2) & \mathbf{v}(3) & \mathbf{v}(1) \end{pmatrix}$$

Then, it can be readily checked that

$$\psi_{\circledast}^{-1}(\psi_{\circledast}(\mathbf{v}_{\sigma_1}) \ldots \psi_{\circledast}(\mathbf{v}_{\sigma_k})) = \mathbf{v}_{\sigma_1} \circledast \ldots \circledast \mathbf{v}_{\sigma_k} = \mathbf{v}_w.$$

4.2.4.4 Permutation-based Approaches

Sahlgren et al. [2008] use permutations on vectors to account for word order. In this approach, given a token σ_m occurring in a sentence $w = \sigma_1 \ldots \sigma_k$ with predefined "uncontextualized" vectors $\mathbf{v}_{\sigma_1} \ldots \mathbf{v}_{\sigma_k}$, we compute the contextualized vector $\mathbf{v}_{w,m}$ for σ_m by

$$\mathbf{v}_{w,m} = \Phi^{1-m}(\mathbf{v}_{\sigma_1}) + \ldots + \Phi^{k-m}(\mathbf{v}_{\sigma_k}),$$

which can be equivalently transformed into

$$\Phi^{1-m}\big(\mathbf{v}_{\sigma_1} + \Phi(\ldots + \Phi(\mathbf{v}_{\sigma_{k-1}} + (\Phi(\mathbf{v}_{\sigma_k}))) \ldots)\big).$$

Note that the approach is still token-centered, i.e., a vector representation of a token is endowed with contextual representations of surrounding tokens. Nevertheless, this setting can be transferred to a CMSM setting by recording the position of the focused token as an additional parameter. Now, by assigning every $\mathbf{v}_\sigma$ the matrix

$$\psi_\Phi(\mathbf{v}_\sigma) = \left(\begin{array}{c|c} & 0 \\ M_\Phi & \vdots \\ & 0 \\ \hline \mathbf{v}_\sigma & 1 \end{array}\right)$$

we observe that for

$$M_{w,m} := (M_\Phi^-)^{m-1}\psi_\Phi(\mathbf{v}_{\sigma_1})\ldots\psi_\Phi(\mathbf{v}_{\sigma_k})$$

we have

$$M_{w,m} = \left(\begin{array}{c|c} \multirow{3}{*}{M_Φ^{k-m}} & 0 \\ & \vdots \\ & 0 \\ \hline \mathbf{v}_{w,m} & 1 \end{array}\right),$$

hence $\psi_\Phi^{-1}\big((M_\Phi^-)^{m-1}\psi_\Phi(\mathbf{v}_{\sigma_1})\ldots\psi_\Phi(\mathbf{v}_{\sigma_k})\big) = \mathbf{v}_{w,m}$.

4.2.5 CMSMs Encode Symbolic Approaches

Now we will elaborate on symbolic approaches to language, i.e., discrete grammar formalisms, and show how they can conveniently be embedded into CMSMs. This might come as a surprise, as the apparent likeness of CMSMs to vector-space models may suggest incompatibility to discrete settings.

4.2.5.1 Group Theory

Group theory and grammar formalisms based on groups and pre-groups play an important role in computational linguistics [Dymetman, 1998; Lambek, 1958]. From the perspective of our compositionality framework, those approaches employ a group (or pre-group) $(G, \cdot)$ as semantical space $\mathbb{S}$ where the group operation (often written as multiplication) is used as composition operation $\bowtie$.

According to Cayley's Theorem [Cayley, 1854], every group G is isomorphic to a permutation group on some set S. Hence, assuming finiteness of G and consequently S, we can encode group-based grammar formalisms into CMSMs in a straightforward way by using permutation matrices of size $|S| \times |S|$.

4.2.5.2 Regular Languages

Regular languages constitute a basic type of languages characterized by a symbolic formalism. We will show how to select the assignment $[\![\ \cdot\ ]\!]$ for a CMSM such that the matrix associated to a token sequence exhibits whether this sequence belongs to a given regular language, that is if it is accepted by a given finite state automaton. As usual (cf. e.g., Hopcroft and Ullman [1979]) we define a nondeterministic finite automaton

$\mathcal{A} = (Q, \Sigma, \Delta, Q_\mathrm{I}, Q_\mathrm{F})$ with $Q = \{q_0, \ldots, q_{n-1}\}$ being the set of states, Σ the input alphabet, $\Delta \subseteq Q \times \Sigma \times Q$ the transition relation, and Q_I and Q_F being the sets of initial and final states, respectively.

Then we assign to every token $\sigma \in \Sigma$ the $n \times n$ matrix $[\![\sigma]\!] = M$ with

$$M(i,j) = \begin{cases} 1 \text{ if } (q_i, \sigma, q_j) \in \Delta, \\ 0 \text{ otherwise.} \end{cases}$$

Hence essentially, the matrix M encodes all state transitions which can be caused by the input σ. Likewise, for a word $w = \sigma_1 \ldots \sigma_k \in \Sigma^*$, the matrix $M_w := [\![\sigma_1]\!] \ldots [\![\sigma_k]\!]$ will encode all state transitions mediated by w. Finally, if we define vectors $\mathbf{v}_\mathrm{I}$ and $\mathbf{v}_\mathrm{F}$ by

$$\mathbf{v}_\mathrm{I}(i) = \begin{cases} 1 \text{ if } q_i \in Q_\mathrm{I}, \\ 0 \text{ otherwise,} \end{cases} \quad \mathbf{v}_\mathrm{F}(i) = \begin{cases} 1 \text{ if } q_i \in Q_\mathrm{F}, \\ 0 \text{ otherwise,} \end{cases}$$

then we find that w is accepted by $\mathcal{A}$ exactly if $\mathbf{v}_\mathrm{I} M_w \mathbf{v}_\mathrm{F}^T \geq 1$.

4.2.5.3 The General Case: Matrix Grammars

Motivated by the above findings, we now define a general notion of matrix grammars as follows:

Definition 1 *Let Σ be an alphabet. A* matrix grammar *$\mathcal{M}$ of degree n is defined as the pair $\langle\ [\![\ \cdot\ ]\!],\ AC\rangle$ where $[\![\ \cdot\ ]\!]$ is a mapping from Σ to $n \times n$ matrices and $AC = \{\langle \mathbf{v}'_1, \mathbf{v}_1, r_1\rangle, \ldots, \langle \mathbf{v}'_m, \mathbf{v}_m, r_m\rangle\}$ with $\mathbf{v}'_1, \mathbf{v}_1, \ldots, \mathbf{v}'_m, \mathbf{v}_m \in \mathbb{R}^n$ and $r_1, \ldots, r_m \in \mathbb{R}$ is a finite set of* acceptance conditions*. The language generated by $\mathcal{M}$ (denoted by $L(\mathcal{M})$) contains a token sequence $\sigma_1 \ldots \sigma_k \in \Sigma^*$ exactly if $\mathbf{v}'_i [\![\sigma_1]\!] \ldots [\![\sigma_k]\!] \mathbf{v}_i^T \geq r_i$ for all $i \in \{1, \ldots, m\}$. We will call a language L* matricible *if $L = L(\mathcal{M})$ for some matrix grammar $\mathcal{M}$.*

Then, the following proposition is a direct consequence from the preceding section.

Proposition 1 *Regular languages are matricible.*

However, as demonstrated by the subsequent examples, also many non-regular and even non-context-free languages are matricible, hinting at the expressivity of our grammar model.

Example 1 *We define* $\mathcal{M}\langle\ [\![\ \cdot\ ]\!],\ AC\rangle$ *with*

$$\Sigma = \{a, b, c\} \qquad [\![a]\!] = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[\![b]\!] = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \qquad [\![c]\!] = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} AC = \{\ & \langle(0\quad 0\quad 1\quad 1), (1\quad -1\quad 0\quad 0), 0\rangle, \\ & \langle(0\quad 0\quad 1\quad 1), (-1\quad 1\quad 0\quad 0), 0\rangle\} \end{aligned}$$

Then $L(\mathcal{M})$ *contains exactly all palindromes from* $\{a, b, c\}^*$*, i.e., the words* $d_1d_2 \dots d_{n-1}d_n$ *for which* $d_1d_2 \dots d_{n-1}d_n = d_nd_{n-1} \dots d_2d_1$.

Example 2 *We define* $\mathcal{M} = \langle\ [\![\ \cdot\ ]\!],\ AC\rangle$ *with*

$$\Sigma = \{a, b, c\} \qquad [\![a]\!] = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[\![b]\!] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad [\![c]\!] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\begin{aligned} AC = \{\ & \langle(1\quad 0\quad 0\quad 0\quad 0\quad 0), (0\quad 0\quad 1\quad 0\quad 0\quad 0), 1\rangle, \\ & \langle(0\quad 0\quad 0\quad 1\quad 1\quad 0), (0\quad 0\quad 0\quad 1\quad -1\quad 0), 0\rangle, \\ & \langle(0\quad 0\quad 0\quad 0\quad 1\quad 1), (0\quad 0\quad 0\quad 0\quad 1\quad -1), 0\rangle, \\ & \langle(0\quad 0\quad 0\quad 1\quad 1\quad 0), (0\quad 0\quad 0\quad -1\quad 0\quad 1), 0\rangle\} \end{aligned}$$

Then $L(\mathcal{M})$ *is the (non-context-free) language* $\{a^mb^mc^m \mid m > 0\}$.

The following properties of matrix grammars and matricible language are straightforward.

Proposition 2 *All languages characterized by a set of linear equations on the letter counts are matricible.*

Proof. Suppose $\Sigma = \{a_1, \ldots a_n\}$. Given a word w, let x_i denote the number of occurrences of a_i in w. A linear equation on the letter counts has the form

$$k_1 x_1 + \ldots + k_n x_n = k \qquad \big(k, k_1, \ldots, k_n \in \mathbb{R}\big)$$

Now define $[\![a_i]\!] = \psi_+(\mathbf{e}_i)$, where $\mathbf{e}_i$ is the ith unit vector, i.e. it contains a 1 at he ith position and 0 in all other positions. Then, it is easy to see that w will be mapped to $M = \psi_+(x_1 \quad \cdots \quad x_n)$. Due to the fact that $\mathbf{e}_{n+1}M = (x_1 \quad \cdots \quad x_n \quad 1)$ we can enforce the above linear equation by defining the acceptance conditions

$$\begin{aligned} AC = \{ & \langle \mathbf{e}_{n+1}, (k_1 \quad \ldots \quad k_n \quad -k), 0\rangle, \\ & \langle -\mathbf{e}_{n+1}, (k_1 \quad \ldots \quad k_n \quad -k), 0\rangle\}. \end{aligned}$$

q.e.d.

Proposition 3 *The intersection of two matricible languages is again a matricible language.*

Proof. This is a direct consequence of the considerations in Section 4.2.6 together with the observation, that the new set of acceptance conditions is trivially obtained from the old ones with adapted dimensionalities. *q.e.d.*

Note that the fact that the language $\{a^m b^m c^m \mid m > 0\}$ is matricible, as demonstrated in Example 2 is a straightforward consequence of the Propositions 1, 2, and 3, since the language in question can be described as the intersection of the regular language $a^+b^+c^+$ with the language characterized by the equations $x_a - x_b = 0$ and $x_b - x_c = 0$. We proceed by giving another account of the expressivity of matrix grammars by showing undecidability of the emptiness problem.

Proposition 4 *The problem whether there is a word which is accepted by a given matrix grammar is undecidable.*

Proof. The undecidable *Post correspondence problem* [Post, 1946] is described as follows: given two lists of words $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ over some alphabet Σ', is there a sequence of numbers $h_1, \ldots, h_m$ $(1 \leq h_j \leq n)$ such that $u_{h_1} \ldots u_{h_m} = v_{h_1} \ldots v_{h_m}$?

We now reduce this problem to the emptiness problem of a matrix grammar. W.l.o.g., let $\Sigma' = \{a_1, \ldots, a_k\}$. We define a bijection $\#$ from Σ'^* to $\mathbb{N}$ by

$$\#(a_{n_1} a_{n_2} \ldots a_{n_l}) = \sum_{i=1}^{l} (n_i - 1) \cdot k^{(l-i)}$$

Note that this is indeed a bijection and that for $w_1, w_2 \in \Sigma'^*$, we have

$$\#(w_1 w_2) = \#(w_1) \cdot k^{|w_2|} + \#(w_2).$$

Now, we define $\mathcal{M}$ as follows:

$$\Sigma = \{b_1, \ldots b_n\} \qquad \llbracket b_i \rrbracket = \begin{pmatrix} k^{|u_i|} & 0 & 0 \\ 0 & k^{|v_i|} & 0 \\ \#(u_i) & \#(v_i) & 1 \end{pmatrix}$$

$$\begin{aligned} AC = \{ \ & \langle (0 \quad 0 \quad 1), (1 \quad -1 \quad 0), 0 \rangle, \\ & \langle (0 \quad 0 \quad 1), (-1 \quad 1 \quad 0), 0 \rangle \} \end{aligned}$$

Using the above fact about $\#$ and a simple induction on m, we find that

$$\llbracket a_{h_1} \rrbracket \ldots \llbracket a_{h_m} \rrbracket = \begin{pmatrix} k^{|u_{h_1} .. u_{h_m}|} & 0 & 0 \\ 0 & k^{|v_{h_1} .. v_{h_m}|} & 0 \\ \#(u_{h_1} \ldots u_{h_m}) & \#(v_{h_1} \ldots v_{h_m}) & 1 \end{pmatrix}$$

Evaluating the two acceptance conditions, we find them satisfied exactly if $\#(u_{h_1} \ldots u_{h_m}) = \#(v_{h_1} \ldots v_{h_m})$. Since $\#$ is a bijection, this is the case if and only if $u_{h_1} \ldots u_{h_m} = v_{h_1} \ldots v_{h_m}$. Therefore $\mathcal{M}$ accepts $b_{h_1} \ldots b_{h_m}$ exactly if the sequence $h_1, \ldots, h_m$ is a solution to the given Post Correspondence Problem. Consequently, the question whether such a solution exists is equivalent to the question whether the language $L(\mathcal{M})$ is non-empty. *q.e.d.*

These results demonstrate that matrix grammars cover a wide range of formal languages. Nevertheless some important questions remain open and need to be clarified next:

Are all context-free languages matricible? We conjecture that this is not the case.[1] Note that this question is directly related to the question whether Lambek calculus can be modeled by matrix grammars.

[1] For instance, we have not been able to find a matrix grammar that recognizes the language of all well-formed parenthesis expressions.

Are matricible languages closed under concatenation? That is: given two arbitrary matricible languages L_1, L_2, is the language $L = \{w_1 w_2 \mid w_1 \in L_1, w_2 \in L_2\}$ again matricible? Being a property common to all language types from the Chomsky hierarchy, answering this question is surprisingly non-trivial for matrix grammars.

In case of a negative answer to one of the above questions it might be worthwhile to introduce an extended notion of context grammars to accommodate those desirable properties. For example, allowing for some nondeterminism by associating several matrices to one token would ensure closure under concatenation.

How do the theoretical properties of matrix grammars depend on the underlying algebraic structure? Remember that we considered matrices containing real numbers as entries. In general, matrices can be defined on top of any mathematical structure that is (at least) a semiring [Golan, 1992]. Examples for semirings are the natural numbers, boolean algebras, or polynomials with natural number coefficients. Therefore, it would be interesting to investigate the influence of the choice of the underlying semiring on the properties of the matrix grammars – possibly non-standard structures turn out to be more appropriate for capturing certain compositional language properties.

4.2.6 Combination of Different Approaches

Another central advantage of the proposed matrix-based models for word meaning is that several matrix models can be easily combined into one. Again assume a sequence $w = \sigma_1 \ldots \sigma_k$ of tokens with associated matrices $\llbracket\sigma_1\rrbracket, \ldots, \llbracket\sigma_k\rrbracket$ according to one specific model and matrices $\llparenthesis\sigma_1\rrparenthesis, \ldots, \llparenthesis\sigma_k\rrparenthesis$ according to another.

Then we can combine the two models into one $\{\![\ \cdot\ ]\!\}$ by assigning to σ_i the matrix

$$\{\![\sigma_i]\!\} = \left(\begin{array}{ccc|ccc} & & & 0 & \cdots & 0 \\ & \llbracket\sigma_i\rrbracket & & \vdots & \ddots & \\ & & & 0 & & 0 \\ \hline 0 & \cdots & 0 & & & \\ \vdots & \ddots & & & \llparenthesis\sigma_i\rrparenthesis & \\ 0 & & 0 & & & \end{array}\right)$$

By doing so, we obtain the correspondence

$$\{\![\sigma_1]\!\}\dots\{\![\sigma_k]\!\} = \left(\begin{array}{c|c} \llbracket\sigma_1\rrbracket\dots\llbracket\sigma_k\rrbracket & \begin{matrix}0&\cdots&0\\ \vdots&\ddots&\\ 0&&0\end{matrix} \\ \hline \begin{matrix}0&\cdots&0\\ \vdots&\ddots&\\ 0&&0\end{matrix} & (\![\sigma_1]\!)\dots(\![\sigma_k]\!) \end{array}\right)$$

In other words, the semantic compositions belonging to two CMSMs can be executed "in parallel." Mark that by providing non-zero entries for the upper right and lower left matrix part, information exchange between the two models can be easily realized.

Hence, we have shown that CMSM is not only algebraically, neurologically and psychologically plausible, but also subsumes the most widespread vector composition operations suggested in other works (see Chapter 3).

5

Evaluation Procedure

In the next chapters of the thesis, we evaluate the Distributional Tensor Space Model presented in 4.1 on a number of standard data sets that are typically used for measuring the quality of distributional models. Further, two novel benchmarks have been suggested. The latter were offered at the workshops that were (partially) co-organized by the author of this thesis.

5.1 Datasets

The enumeration of the evaluated datasets is given here for an overview; the detailed description follows in the corresponding sections. The datasets include:

1. **Free Word Associations**: a shared task from the ESSLLI'2008 Workshop[1];

2. **Similarity Judgements** [Rubenstein, 1965]: estimation of *attributional similarity*;

3. **Selectional Preferences or Thematic Fit** [Bicknell et al., 2010; Lenci, 2011]: tendency for certain words or word categories (grammatical or semantic) to cooccur with certain other words or categories;

4. **Multiword Units (MWU)** [Evert and Krenn, 2001; Katz and Giesbrecht, 2006]: automatic classification of MWU in compositional, that is literal, or non-compositional, that is, figurative;

[1] http://wordspace.collocations.de/doku.php/workshop:esslli:task

5. **DiSCo Shared Task** [Biemann and Giesbrecht, 2011]: phrase compositionality detection, containing *adjective-noun*, *verb-object* and *subject-verb* constructions;

6. **SemEval 2013, Task 5b** [Korkontzelos et al., 2013]: distinguishing between figurative and literal usages of a phrase in context;

7. **Phrase Similarity** Dataset [Mitchell and Lapata, 2010]: containing *adjective-noun*, *verb-object* and *noun-noun* pairs;

8. **Transitive Sentence Similarity** Task [Grefenstette and Sadrzadeh, 2011b]: predicting similarity for sentences consisting of *subjects*, *verbs* and *direct objects*.

5.2 Evaluation Metrics

The measures used for the evaluation of the tasks include the following:

- **"Information Retrieval" measures**

  Accuracy: $$\frac{true\_positives + true\_negatives}{true\_positives + true\_negatives + false\_positives + false\_negatives}$$

  Precision (P): $$P = \frac{true\_positives}{true\_positives + false\_positives}$$

  Recall (R): $$R = \frac{true\_positives}{true\_positives + false\_negatives}$$

  F-measure (F): $$F_1 = 2 \times \frac{P \times R}{P + R}$$

- **Statistical correlations**

  Correlation measures the degree of relationship between two variables. It can take values between -1 and 1. 0 indicates no relationship between the variables. 1 or -1 indicates a linear relationships, such that if one variable is known, the second can be accurately predicted. Positive correlation shows that if one variable increases, the other should grow either. A negative coefficient means that if one variable increases, the other decreases. **Pearson's** r and **Spearman's** ρ are the most widespread correlation measures.

Pearson's r quantifies a linear relationship between given numbers (X_i and Y_i), if any exists.

$$r = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_i (X_i - \overline{X})^2 (Y_i - \overline{Y})^2}}$$

Spearman's ρ is used for measuring rank correlations; that is, the resulting scores X_i and Y_i are transformed to ranks x_i and y_i and then the correlation is computed in a similar way as for the Pearson coefficient.

$$\rho = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2 (y_i - \overline{y})^2}}$$

In order to find the correlation or compute the accuracy, we need reference data that present the **"truth"**.

Gold Standard is an important notion in any evaluation exercise. It is usually a manually constructed benchmark that is used to measure the performance of the automatic algorithm. Ideally at least two human annotators construct such a dataset and the level of agreement between the annotators is called **inter-annotator agreement**. The latter is considered to be "the upper bound" for algorithm's performance.

5.3 Computational Resources and Frameworks

All experiments in this thesis, concerning Distributional Tensor Space Model, were conducted on Red Hat 4.1.2 server operating system running on Intel Core 4 Xeon @ 2.33 GHz CPU with 10 GB RAM; the task of detecting non-compositionality for multiword units (cf. Section 7.1) as well as the task of measuring free words associations (cf. Section 6.1) were performed on a regular 2.53 GHz Intel Core i5 laptop with 8 GB RAM.

Text corpora used for evaluation include the following:

- UK Web as Corpus (ukWaC) [Ferraresi et al., 2008] - for free word associations;
- an excerpt of Süddeutsche Zeitung (SZ) corpus for 2003 - for multiword units detection;

- British National Corpus (BNC[1]).

  The majority of experiments in this thesis are based on the British National Corpus (BNC), a 100 million word collection of written and spoken English collected from a wide range of sources. The corpus contains 100,106,008 words in 4124 texts consisting of 6.25 million sentences. The corpus is lemmatized and part-of-speech tagged.

Following **Tools** have been used or implemented for this work:

- Semantic Vectors package [Widdows and Ferraro, 2008];
- MATLAB Tensor Toolbox for tensor decomposition [Bader et al., 2012];

  The Tensor Toolbox supports operations for sparse tensors that we deal with. We use a sparse implementation of Tucker, PARAFAC and NMF algorithms.
- Self-implemented tensor manipulation framework in Java, using frameworks and libraries Spring[2], Hibernate[3], Colt[4] and Ant[5].

5.4 Construction of Distributional Tensor Space Model

For all of the experiments, except for one concerning multiword units (Section 7.1), we proceed in the following way:

1. build a Distributional Tensor Space Model from one of the above listed corpora;

   For this, we possibly fix in advance a number of parameters (context window size, number and choice of context dimensions, number of factors for decomposition) or experiment with those.
2. extract either vectors (tensor fibers) or matrices (tensor slices) representing words from the resulting $3d$ tensor;

[1] http://www.natcorp.ox.ac.uk

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

[3] http://www.hibernate.org/

[4] http://acs.lbl.gov/software/colt/

[5] http://ant.apache.org/

3. for phrases and sentences: compose component word matrices by means of matrix addition and multiplication;
4. compute cosine similarity for extracted or composed vectors and matrices;
5. calculate either precision or correlations for the resulting similarity scores.

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

In the traditions of distributional models, we avoid the so-called **stop words** to be included into the model. Below is an example of a stop word list that was used:

```
a, able, about, across, after, all, almost, also, am, among,
an, and, any, are, as, at, be, because, been, but, by, can,
cannot, could, dear, did, do, does, either, else, ever,
every, for, from, get, got, had, has, have, he, her, hers,
him, his, how, however, i, if, in, into, is, it, its, just,
least, let, like, likely, may, me, might, most, must, my,
neither, no, nor, not, of, off, often, on, only, or, other,
our, own, rather, said, say, says, she, should, since, so,
some, than, that, the, their, them, then, there, these,
they, this, tis, to, too, twas, us, wants, was, we, were,
what, when, where, which, while, who, whom, why, will, with,
would, yet, you, your
```

5.5 Tensor Exploitation

A tensor allows a number of usages.

Having three axes, we can decide which of the axis should be used as a reference point for a word matrix initialization. Consequently, we can extract the X, Y or Z slices of the tensor that refer to different kinds of contexts (see Figures 5.1, 5.2, 5.3). In the general case, we will use the Y slice of the tensor which corresponds to the middle context words; thereby the corresponding matrix contains the left and the right contexts of the target concept (cf. Experiments in Chapter 6 (6.1 and 6.2) and Chapter 7 (7.2, 7.3, 7.4 and 7.5)).

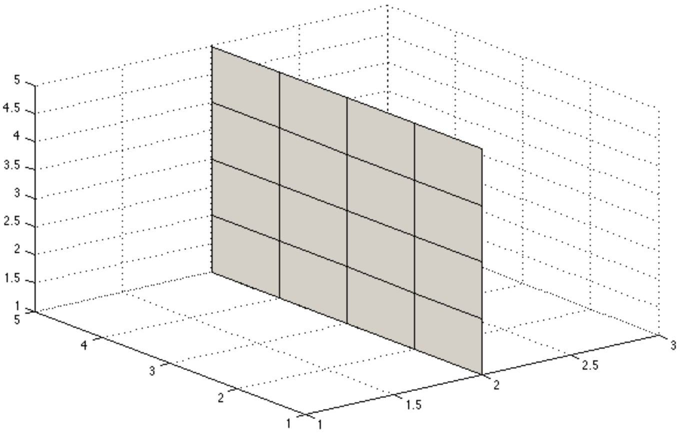

Figure 5.1: X-Slice of Tensor

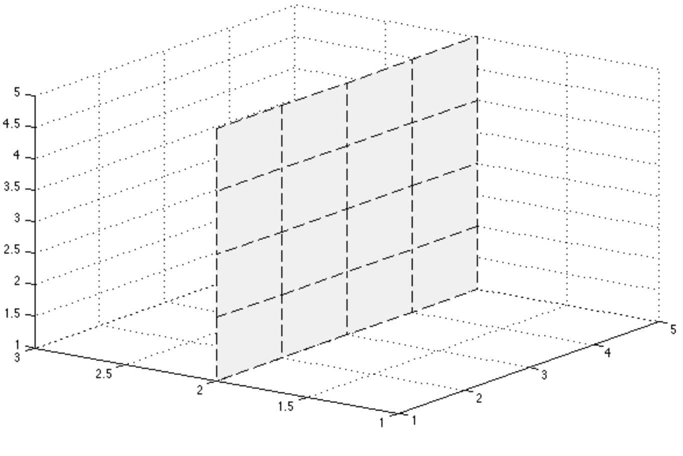

Figure 5.2: Y-Slice of Tensor

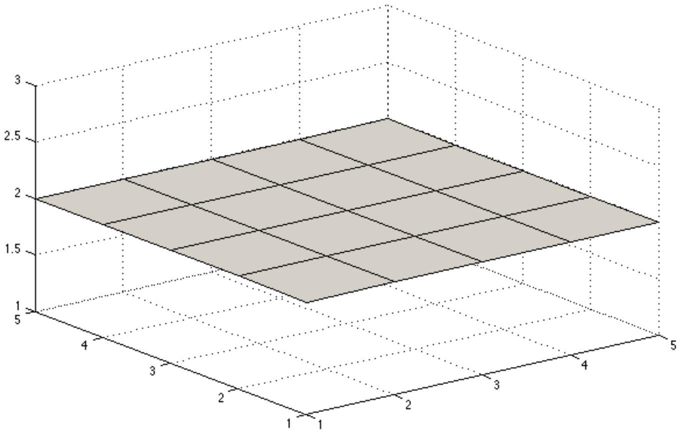

Figure 5.3: Z-Slice of Tensor

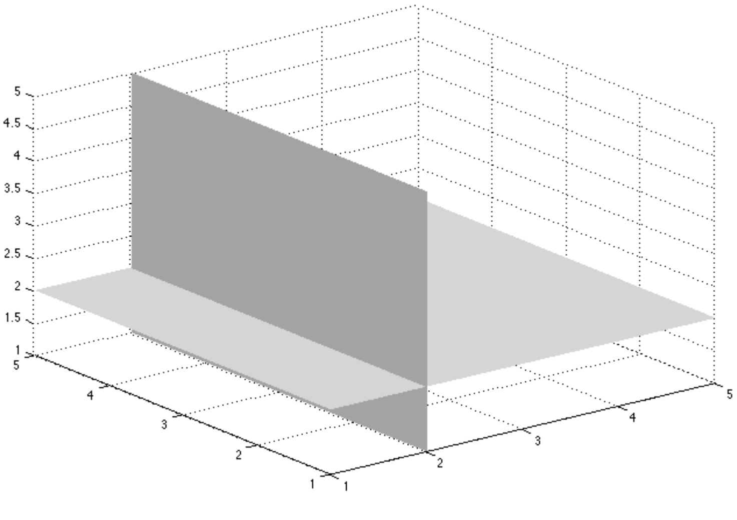

Figure 5.4: Intersection of two slices: row vector, or fiber, extraction

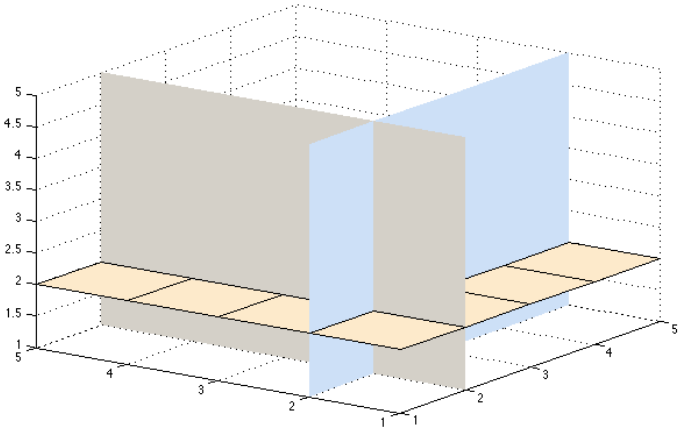

Figure 5.5: Intersection of three slices: finding point values

Further, in some cases we may need to extract a vector at the intersection of two slices, i.e., matrices (see Figure 5.4).

The third way to deploy the tensor is to find a concrete point at the intersection of three slices (cf. Figure 5.5). This methodology is relevant, for example, for the exercise of determining selectional preferences (Chapter 6 (6.3)).

6

Experimental Part I: Model Evaluation

This chapter describes the evaluation of the proposed model on a number of datasets that have established themselves as benchmarks for measuring the performance of semantic space models.

Traditionally, distributional semantics methods have been used for a number of tasks on automatic discovery of *semantic relatedness* between words, like TOEFL synonymy test [e.g. Rapp, 2003] or detection of analogical similarity [e.g. Turney, 2006].

The performance of corpus-based state-of-the-art methods[1] for TOEFL synonymy task has achieved in the meantime the perfection of 100% [Bullinaria and Levy, 2012], which is even above a possible human performance on this task; and there is nothing more that can be done in respect to this task.

A related exercise that turned out to be much more sophisticated is the task of finding out to what extent (statistical) similarity measures correlate with free word associations (see Section 6.1).

Another widespread benchmark is the dataset of Rubenstein [1965] (see Section 6.2) where graded similarity values were assigned to the pairs of concepts.

[1] http://aclweb.org/aclwiki/index.php?title=TOEFL_Synonym_Questions_(State_of_the_art)

An important task in many practical applications is the task of selectional preferences. The latter play significant role in many application areas, such as question answering, ontology construction, query answering, paraphrasing; in computational linguistics research - for parsing, word-sense disambiguation, metaphor recognition; in cognitive science - for understanding of the organization of mental lexicon and knowledge in the brains. Following Lenci [2011], we use the dataset of Bicknell et al. [2010] to evaluate DTSM on the task of selectional preferences.

In the following, we describe the corresponding datasets and present the evaluation results for the Distributional Tensor Space Model.

6.1 Free Word Associations Task

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.* Examples of free associations are: MATTRESS - BED, REFLECTION - MIRROR, CAT - DOG, etc. These associations do not correspond to just *one* kind of semantic relation, that is why they are called *free.* The latter makes their computational analysis difficult as there is up-to-now no unified computational theory for analysing all possible kinds of semantic relations and new models are built for every task.

The *free word association* task was suggested as a *shared task* for the evaluation of word space modelsat Lexical Semantics Workshop at ESSLLI 2008, and it is freely available[1]. For this task, workshop organizers proposed three subtasks, one of which - *discrimination* - we evaluate here.

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 (e.g., GIRL - BOY);

[1] http://wordspace.collocations.de/doku.php/workshop:esslli:task

- HAPAX - word associations that were produced by a single test subject (e.g., CAFE - FISH);
- RANDOM - random combinations of words (e.g., DIGITAL - REVOLT).

To collect the three-way co-occurrence information, we experiment with the UKWAC corpus 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 to process the whole corpus. As the sub-sections of UKWAC contain randomly chosen documents, one can train the model on any of the sub-sections.

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.

6.1.1 Procedure

We proceed in the following way - for *each pair of words*:

1. gather N-sentences for each of the two component words;
2. build a 3-dimensional tensor from the subcorpus obtained in (1), given a window size w (here: $w = 10$, i.e. 5 words to the left and 5 words to the right of the target word);
3. reduce 5 times the dimensionality of the tensor obtained in (2) by means of Tucker decomposition using Matlab Tensor Toolbox Bader and Kolda [2006];
4. extract two matrices of both constituents of the word pair and compare those by means of cosine similarity.

6.1.2 Results

Tables 6.1 and 6.2 show the resulting accuracies for the training and test sets. th denotes cosine threshold values that are used for classification of the results. Here, th is taken to be the function of the dataset size. Thus, given a training set of size $s = 60$ and 3 classes, de define an "equally distributed" threshold

$th_1 = 60/3 = 20$ (Table 6.1) and a "linearly growing" threshold $th_2 = \frac{1}{4}, \frac{1}{3}, rest$ (Table 6.2).

It is not apparent yet, how and if a "universal" threshold for differentiating between the groups should and can be determined. The results from Tables 6.1 and 6.2 show that the final cosine similarity threshold is dependent on the corpus and experiment settings. Maybe, in the same way as the measure of similarity cannot be easily defined by humans, so the threshold for geometric models of meaning cannot be just fixed.

	TRAIN	TEST
FIRST	12/20 (60%) ($th_1 = 0.022$)	25/74 (33%) ($th_1 = 0.078$)
HAPAX	7/20 (35%) ($th_1 = 0.008$)	35/74 (47%) ($th_1 = 0.042$)
RANDOM	8/20 (40%)	23/74 (31%)
Total (F/H/R)	27/60 (45%)	83/222 (37.4%)
FIRST/HorR[1]	44/60 (73.33%)	125/222 (56.3%)

Table 6.1: Free word associations: Accuracies for *equally distributed* threshold

	TRAIN	TEST
FIRST	9/15 (60%) ($th_2 = 0.0309$)	20/55 (36.4%) ($th_2 = 0.09$)
HAPAX	8/20 (40%) ($th_2 = 0.0101$)	39/74 (52.7%) ($th_2 = 0.047$)
RANDOM	10/25 (40%)	24/93 (25.8%)
Total (F/H/R)	27/60 (45%)	108/222 (48.6%)
FIRST/HorR[2]	43/60 (71.60%)	113/222 (50.9%)

Table 6.2: Free word associations: Accuracies for *linear growing* threshold

In contrast to the analogue LSA-based model that was reported by Wandmacher et al. [2008] to obtain good results for RANDOM associations but the lowest results for the FIRST, i.e. strongest associations, our model was more accurate for the FIRST, i.e. strongly associated, word pairs.

The overall results (ca. 50% to 73% accuracy depending on the task setting) seem not to be quite optimal at this stage. However, the reported results have been obtained based on very small corpora, containing basically 100 sentences per iteration (cf. Wandmacher et al. [2008] use a corpus of 108M words to train their LSA-Model).

Another important issue with *free association norms* is that they are inherently sensitive to quite a number of factors: social, economic, political and cultural background - all influence the kind of associations people make. These are all the factors that need to be considered in the future by choosing the training corpus and test data.

6.2 Similarity Judgements

The dataset of Rubenstein [1965] contains 65 noun pairs that were manually rated for similarity on a 0-4 scale by 51 subjects. The average of these ratings is reported as a similarity value in the dataset: e.g., *car-automobile* has a score of 3.92, *mid and noon* - 3.94 and *noon and string* - 0.04.

Similarly to the others [Padó and Lapata, 2007; Baroni and Lenci, 2010], we use Pearson's r to evaluate the correlation between the scores of our system and the ones from the gold standard. Rubenstein [1965] report inter-annotator correlation of r = 0.85. This score can be thought of as an upper bound for computational methods.

The resulting correlations for words that occurred within collected triples in the BNC corpus more than 5 times are presented in Table 6.3. Obviously, *better results* are achieved with *bigger context window.* Using *more decomposition factors* brings further improvement, especially *with less context dimensions.* Thus, we've achieved the currently best result with *2000 dimensions, 100 factors and max. 13 neighbours per side* taking into consideration sentence boundaries.

Recall is reported in parenthesis in Table 6.3. We take recall to be less important in this case, as we can always achieve it with more data; the latter is obvious from Table 6.4 which shows that the recall of one is achieved for this dataset if we just take all triples into account without penalizing for frequency.

Table 6.5 shows the result of the so far best DTSM model and three further state-of-the-art results; two of which (TypeDM and WIN) are based on much bigger corpora (ukWaC + Wikipedia). DV-cosine is the only method that was tested on the same corpus as our model, i.e. the BNC.

number of dimensions	factors	neighbours per side	Pearson's r (Recall)
5000	50	5	0.29 (0.62)
	50	13	0.42 (0.68)
	100	5	0.44 (0.62)
	100	13	0.43 (0.68)
2000	50	5	0.36 (0.58)
	50	13	0.50 (0.68)
	100	5	0.36 (0.67)
	100	**13**	**0.54** (0.68)

Table 6.3: Rubenstein and Goodenough (1965): Pearson correlations for DTSM, (recall in parenthesis)

DTSM 2000, $f100$	**min1**	**min5**
$nn = 5$	0.49 (1.0)	0.44 (0.585)
$nn = 13$	0.51 (1.0)	0.54 (0.68)

Table 6.4: Rubenstein and Goodenough (1965): Pearson correlations for DTSM$_{best}$ with varying minimum triple occurrences

DTSM	**0.54**
TypeDM [Baroni and Lenci, 2010]	0.82
WIN	0.65
...	...
DV-cosine [Padó and Lapata, 2007]	**0.47**

Table 6.5: Rubenstein and Goodenough (1965): Pearson correlations for DTSM and state-of-the-art models of similar background

For comparability, we mention here only the results of the models that come from the similar line of research, i.e. purely distributional semantics based approaches. TypeDM is here representative for all *distributional memory* group of approaches as the one with the best performance [Baroni and Lenci, 2010].

Pearson's correlation of 0.47 for dependency vector model in Table 6.5 is reported for the model of Padó and Lapata [2007] that uses cosine similarity, 2000 basis elements and the log-likelihood association function. Padó and Lapata [2007] report also a much better correlation of 0.62 which is, however, due to the other similarity measure [Lin, 1998] that was used instead of cosine. As this is one of the tunable parameters that can also be applied to our model, we compare to the setup that is used in our model, i.e., using the same similarity measure, namely the cosine.

DTSM is performing worse than TypeDM; the latter can be explained, first of all, by the corpus size and type - **BNC** versus **ukWaC**, that is, 100 million word collection containing highly standardized newspaper, fiction and similar text genres versus 2 billion word corpus constructed from the Web.

Furthermore, it may be the case that vector-based approaches are more suited than a more sophisticated matrix approach for this type of similarity that is asked for in this dataset, as well as other word similarity tasks where the notion of similarity is rather arbitrary defined, in the sense that it is mostly independent of the structural effects that present the added value of the third dimension that is offered by our model.

6.3 Thematic Fit, or Selectional Preferences

All of the benchmarks, described so far, have been suggested and are widely used for the evaluation of the semantic tasks on the *word level*. In this section we address one of the most important and "unsolved" tasks in NLP - the task of *selectional preferences* - where word order information matters.

The topic of **selectional preferences** (in computational linguistics), also known as **thematic fit** (in (psycho-)linguistics), is an important aspect for human sentence processing [McRae et al., 1998; Lenci, 2011]. On the psycholinguistic side, knowing selectional preferences helps us to tell plausible sentences from implausible ones. In cognitive science, understanding thematic fit reveals insights into human concept formation and mental lexicon organization.

Selectional preferences, or selectional constraints, are restrictions on the applicability and plausibility of certain word combinaitions, e.g., if ”green grass” is absolutely natural, ”green cow” is unlikely but conceivable, ”green idea” is both unlikely and unimaginable [Resnik, 1996]. This topic has long roots in the formal truth-theoretic semantics and there has been a number of approaches for automatic acquisition of such preferences from text, since Resnik [1993, 1996] formalized the first computational model of selectional preferences[1].

Usually, this task is associated with selectional preferences for verbs in computational linguistics, i.e. with automatic extraction of plausible arguments (subject and objects) for a predicate.

In psychology and cognitive science, it has been shown that verbs activate expectations about nouns occurring as their arguments, e.g. when hearing *prepare*, one may think of *dinner* or *speech*, and vice versa, if you remember that you have to give a *speech* tomorrow, you will probably speculate about *preparing* it [McRae et al., 1998, 2005]. Furthermore, nouns also activate expectations in the brains about other nouns occurring as co-arguments in the same event, like *key* - *door* or *politician* - *speech* [Hare et al., 2009].

Bicknell et al. [2010] demonstrate an even more complex view of verb-argument expectations, showing the three-way dependencies between verbs and both of its arguments. For instance, if the agent noun is *chef*, the probable patient for the verb *to prepare* may be, e.g., *dinner*; while if it is a *politician*, then the patient of *prepare* is more likely to be *speech.* Consequently, *thematic fit* is also sensitive to the way other roles of the same verb are filled, i.e., it is *a three-way dependence.*

Lenci [2011] was the first to address the issues of *thematic fit* and *compositionality* by means of distributional semantic models together as one task. For this, he uses Distributional Memory framework (TypeDM, $W_1 \times LW_2$ space), described in Section 3.2.4 to compute expectations of *agent-verb* pairs for corresponding

[1] See, for example, Resnik [1996] and Erk et al. [2010] for an overview of theoretical discussions and computational approaches correspondingly.

patients. The thematic fit of the potential *patient* is measured by the cosine between its TypeDM vector and the "prototype" vector which is obtained out of the top-k objects of the *agent+verb* pair:

$$EX_{PA}(\langle n_{AG}, v\rangle) = f(EX(n_{AG}), EX_{PA}(v)) \qquad (6.3.1)$$

- with f being SUM or $PRODUCT$ and where $EX(n_{AG})$ is the set of TypeDM tuples of the form $\langle n_{AG}, verb, n_j\rangle$ and $EX_{PA}(v)$ is the set of TypeDM tuples of the form $\langle n_i, obj, v\rangle$.

For example, $EX(mechanic)$ could be tuples ⟨*mechanic*, *verb*, *car*⟩, ⟨*mechanic*, *verb*, *oil*⟩, ⟨*mechanic*, *verb*, *engine*⟩ and so on. $EX_{PA}(check)$ could be represented by ⟨*mistake*, *obj*, *check*⟩, ⟨*engine*, *obj*, *check*⟩, etc.

Similar to Lenci [2011], we use the dataset of Bicknell et al. [2010] (**bicknell.64**) to evaluate the performance of our model for selectional preferences. The dataset contains 64 contrastive triples of word pairs, each sharing the same verb, but differing for the agent and patient nouns, e.g.:

```
journalist - check - spelling
  mechanic - check - brake
```

Patients in each triple were produced by 47 subjects as the prototypical (congruent) arguments of the verbs given a certain agent. The patient noun in one triple is incongruent with another triple with the same verb but a different agent: e.g., *brake* is an incongruent patient for the *journalist* - *check* pair but congruent for the *mechanic* - *check* combination.

6.3.1 Procedure

Our Distributional Tensor Space Model allows to extract such dependencies in a completely unsupervised manner from text in a straightforward way.

For all the congruent and incongruent triples, we extract a value from the tensor at the intersection of corresponding X, Y and Z indices for subject, verb and object. For example, for the pair *journalist* - *check* - *spelling* we identify the index of *journalist* on the *X-axis*, the index of *check* on the *Y-axis* and *spelling* on the *Z-axis*. Then the intersection of the three slices results in a number (cf. Figure 5.5). The same is done for the incongruent pair. The triple that gets a bigger number is considered as congruent.

6.3.2 Results

We report here precision values for this experiment. We consider precision to be more important in this case, as recall can be remedied with more data. Compared to Lenci [2011], we use a relatively small corpus (BNC) for this proof of concept.

There are two ways how we can interpret congruency evaluation in this case: taking either *agents* or *patients* as points of reference. Lenci [2011] uses the former (agents); i.e., he measures the precision by means of comparing cosine $(f(\mathbf{journalist}, check), spelling)$ and $cosine(f(\mathbf{mechanic}, check), spelling)$. In this case, if the first is bigger, the answer of the system is interpreted as correct; otherwise fail. Table 6.6 reports the corresponding numbers for our DTSM model.

Rather high precision that we achieve (0.72) is the second best, compared to the results reported in Lenci [2011]. It is worth reminding that we use text collection of the size at most $\frac{1}{10}$th of the corpus used in the original experiment. Furthermore, we do not invent any special way to treat this exercise that is optimized only for the given task; we use the model as it is and achieve very high precision. All of the evaluated settings of DTSM are better than the baseline[1], except for one; and they are better than the second best model of Lenci [2011].

We also report the precision for patients given the same agent, i.e., we compare $cosine(journalist, check, \mathbf{spelling})$ and $cosine(journalist, check, \mathbf{break})$ (cf. Table 6.7). Even better results are achieved for the second verb argument, i.e., patient. However, no results for the *patient argument* were reported in Lenci [2011], so we cannot directly compare our results to any other system in this case except for baseline.

[1] Baseline here is the probability of choosing one of the two available variants, which is 50%

# context words	# factors	# neighbours	Precision
2000	50	5	0.61
		13	0.68
	100	5	0.66
		13	0.55
	150	5	0.65
		13	0.61
5000	50	5	0.44
		13	0.61
	100	5	0.5
		13	0.61
	150	5	0.38
		13	0.61
10000	50	5	0.47
		13	0.57
	100	5	0.63
		13	0.68
	150	5	0.58
		13	**0.72**
Baseline			0.50
Lenci [2011] best model			0.84
Lenci [2011] 2nd best model			0.41

Table 6.6: Precision of DTSM on **bicknell.64** dataset for triples with minOcc=5 and modifying agents

All in one, the results show that the model can successfully predict the *thematic fit*, or *selectional preferences*, for **both** *agent* and *patient* **verb argument positions** without any task-specific optimization.

# context words	# factors	# neighbours	accuracy
2000	50	5	0.66
		13	0.75
	100	5	0.65
		13	0.635
	150	5	0.66
		13	0.68
5000	50	5	0.48
		13	0.57
	100	5	0.57
		13	**0.77**
	150	5	0.65
		13	0.63
10000	50	5	0.6
		13	0.76
	100	5	0.64
		13	0.62
	150	5	0.6
		13	0.70
Baseline			0.50

Table 6.7: Precision of DTSM on **bicknell.64** dataset for triples with minOcc=5 and modifying patients

7

Experimental Part II: Evaluation of Compositionality

This chapter describes the experiments we performed in order to evaluate the Distributional Tensor Space Model in terms of its ability to reproduce semantic compositionality, i.e. the ability to reconstruct the meaning of a phrase from the meanings of its parts. Any NLP system that does semantic processing has to handle the issue of compositionality of natural language.

Compositionality, as we have seen in Chapter 3.3, is a controversial and much discussed issue in language research. In the same way, there is no consensus on the best way to evaluate the models of compositionality as nobody can truely measure this rather abstract concept.

Several ways of evaluation have been suggested. The early approaches started with evaluation of models of compositionality by means of the opposite, i.e. by trying to figure out if we can automatically measure the non-compositionality of certain multiword expressions (MWEs), also called multiword units (MWUs). Multiword expressions are "idiosyncratic interpretations that cross word boundaries". They cause troubles for both semantic and syntactic processing as they cannot be interpreted by means of direct combination of the meanings of the component words [Sag et al., 2002].

Automatic identification or classification of multiword expressions was recognized as an important task in computational linguistics long before the issue of compositionality in distributional semantics came up. Distributional approaches to MWEs started to gain in importance at the beginning of the 2000s [Schone and Jurafsky, 2001; Baldwin et al., 2003].

It has been empirically shown in Katz and Giesbrecht [2006], that vector similarity between distribution vectors associated with a multiword unit as a whole and those associated with its constituent parts can serve as a good measure of the degree to which the multiword unit is (non-)compositional. However, a number of issues were left open in the end; among them, the question if a better mathematical approximation for simulating compositional meaning, other than addition, would improve the algorithm.

The first part of compositionality evaluation builds upon this line of research on multiword units. Section 7.1 describes a continuation of our previous work, started in Katz and Giesbrecht [2006], by testing different mathematical operations, introduced in Section 3.3.1, on the task of multiword unit identification, making use of Random Indexing [Sahlgren, 2005].

Sections 7.2 presents the shared task that was offered at the Distributional Semantics and Compositionality (DiSCo) workshop. It addresses the problem of *graded* instead of *binary* compositionality classification that has been notoriously ignored in the computational semantics community [Bannard et al., 2003; Katz and Giesbrecht, 2006]. The task consists in letting computational systems automatically assign compositionality scores to phrases instead of simply classifying them as compositional or non-compositional. The dataset contains MWUs without sentence contexts.

Due to the huge interest of the community to the original contexts, in which the phrases occurred, as well as in order to evaluate the promising results of Katz and Giesbrecht [2006] on using local contexts, Task 5b[1] at SemEval (Semantic Evaluation) 2013 competition was suggested. SemEval consists of a number of shared tasks for evaluation of computational semantic systems that are organized within a 1-2 day workshop that is usually co-located with one of the major conferences in computational linguistics. SemEval replaced in 2007 the Senseval Word Sense Disambiguation workshop series that had been conducted since 1998.

[1] http://www.cs.york.ac.uk/semeval-2013/

Section 7.3 describes the SemEval shared task which offers a further extension of MWU classification into compositional, i.e. literal, versus non-compositional, i.e. idiomatic, phrases; but this time they are given in context. We evaluate the Distributional Tensor Space Model on this dataset too.

Both of the shared tasks, described in Section 7.2 and in Section 7.3, were initiated and co-organized by the author of this thesis and present extensions of work done in Katz and Giesbrecht [2006] and Giesbrecht [2009] on compositionality detection.

Another way of evaluating compositional word space models of meaning is following the ideas of traditional distributional semantics similarity tasks in that two pairs of phrases or sentences, instead of simple words, are compared for similarity. A number of datasets has been suggested for this exercise, and they are used as benchmarks in the meanwhile. The task is basically to evaluate distributional models of composition in terms of their ability to predict similarity ratings for simple phrases, i.e. a natural extension of word level similarity tasks, like synonymy or associations tasks.

This evaluation setup was suggested by Kintsch [2001]. However, he demonstrated his algorithm only on a few selected examples; the latter was criticized in literature [Frank et al., 2008]. Nevertheless, the idea suggested by Kintsch [2001] was taken up by many researchers later and extended to large and proper evaluation datasets.

The dataset of Mitchell and Lapata [2008, 2010] is one of those (Section 7.4). It consists of pairs of *adjective-noun*, *verb-object* and *compound noun* phrases.

Last but not least, Grefenstette and Sadrzadeh [2011b] assembled a dataset of transitive sentences containing *subjects*, *verbs* and *direct objects*, motivated by the methodology of Mitchell and Lapata [2010]. We take a chance to test DTSM also on this dataset in the last step (Section 7.5).

7.1 Non-Compositionality Detection for Multiword Units

A multiword unit (MWU) is defined here as a connected sequence of neighbouring words whose exact and unambiguous meaning cannot be derived from the meaning of its components [Schone and Jurafsky, 2001]. *Spill the beans* or *hot dog* are examples of such MWUs. Therefore, a MWU either has a completely opaque meaning, or its constituent words acquire some other nuance of meaning when they are used together, thereby making the expression as a whole non-compositional. Thus, *spill* co-occurring with *the beans* has nothing to do with *slopping* but rather with *revealing (secrets).* In contrast, *buy a ticket* is about *buying* and *ticket* together, and it is perfectly compositional. Figurative, i.e. non-compositional, MWUs have always posted a problem for compositional theories of language and have been used as an objection to the principle of compositionality of human language within symbolic approaches.

There is a long-living tradition within the research community working on multiword units to automatically identify MWUs in text corpora using statistical association measures [Evert and Krenn, 2001; Evert, 2004; Lin, 1999] or by means of Latent Semantic Analysis [Schone and Jurafsky, 2001; Baldwin et al., 2003; Katz and Giesbrecht, 2006].

Schone and Jurafsky [2001] and Katz and Giesbrecht [2006] explored detection of non-compositional phrases by means of comparing the co-occurrence signatures of a multiword unit as a whole and those of the composed vectors of its constituents. The main assumption in all similar experiments is that compositional MWUs appear systematically in contexts more similar to those in which their component words appear than do non-compositional MWUs.

Figure 7.1 illustrates such a vector space in two dimensions. Note that the meaning vector for the MWU *yellow press* is quite similar to that for *gossip* but distant from *yellow*, while the meaning vector for *yellow banana* would be much closer to *yellow* in contrast. Indeed *yellow press* is a non-compositional idiom meaning 'newspapers that publish gossip about celebrities'.

Katz and Giesbrecht [2006] showed that the local context of a MWU could reliably distinguish idiomatic uses of MWU from non-idiomatic uses. It was shown that LSA vectors for compositional and non-compositional uses of an idiom (manually annotated) were orthogonal, i.e., unrelated.

However, both of the above mentioned works define the estimated compositional meaning vector by taking it to be the sum of the component vectors, i.e., the

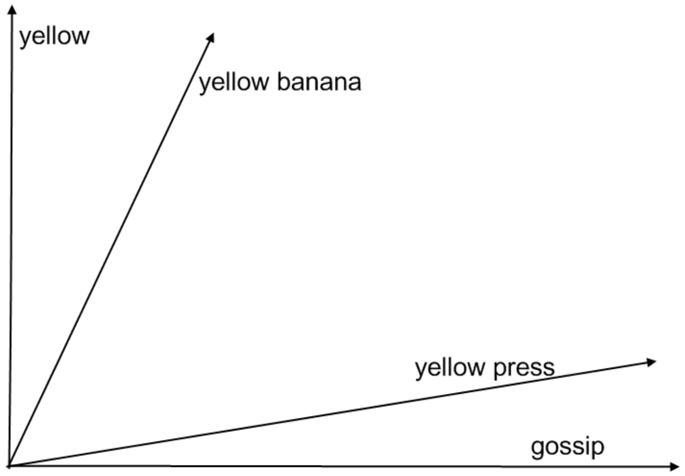

Figure 7.1: Two-dimensional word space

compositional meaning of the expression *yellow banana* is taken to be the sum of the vectors for the corresponding words $vector(yellow) + vector(banana)$. They recognize that the composed vector is clearly nowhere near a perfect model of compositional meaning, but it proved to be just enough to test the hypothesis.

We build here upon the work of Katz and Giesbrecht [2006] and explore more advanced mathematical operations on vectors, suggested by Widdows [2008], as an approximation of "semantic composition" by adopting their evaluation paradigm. In particular, we are looking for an answer to the question whether simply applying more advanced mathematical operations on vectors would be enough to achieve better models of the semantic compositionality in vector spaces.

7.1.1 Compositional Models

Let *w1w2* denote the composition of two vectors *w1* and *w2*. In the following, we define the operations for vector compositionality models that we test later. The estimated compositional meaning vector *w1w2* is calculated by taking it to be:

1. (+) the sum of the meaning vectors of the parts, i.e., the compositional meaning of an expression *w1w2* consisting of two words is taken to be sum

of the meaning vectors for the constituent words *w1* and *w2*: $(w1w2)_i = w1_i + w2_i$;

Thus, the "compositional" vector for *yellow press* in this case would be the sum of the vectors for *yellow* and *press*.

2. $(\cdot)$ the simplified multiplicative model as it is defined in Mitchell and Lapata [Mitchell and Lapata, 2008]: under the assumption that only the ith component of *w1* and *w2* contribute to the ith component of *w1w2*, we can formulate vector multiplication operation as: $(w1w2)_i = w1_i \cdot w2_i$;

3. $(\otimes)$ the tensor product: if the vector of the word *w1* has components $w1_i$ and the vector of the word *w2* has components $w2_j$, then the tensor product $(w1 \otimes w2)$ is a matrix whose ij^{th} entry is $w1_i w2_j$ [cf. Widdows, 2008];

4. $(*)$ the convolution product, which is also a kind of vector multiplication that results in the third vector of dimensionality $(m + n - 1)$. Given two vectors $w1 = [w1_1, w1_2, w1_{...}, w1_m]$ and $w2 = [w2_1, w2_2, w2_{...}, w2_n]$, their convolution $(w1 * w2)$ is defined as $(w1w2)_i = \sum_j w1_j w2_{i-j+1}$.

For computing meaning similarity for vector addition, component multiplication and convolution, we use the standard measure of cosine of the angle between two vectors (the normalized correlation coefficient) as a metric [Schütze, 1998; Baeza-Yates and Ribeiro-Neto, 1999], which corresponds for normalized unit vectors to a scalar product of those. In this metric, two expressions are taken to be unrelated if their meaning vectors are orthogonal (the cosine is 0) and synonymous if their vectors are parallel (the cosine is 1). For the tensor product, the natural similarity measure is the inner product on tensors and is defined as the product of the similarities of the constituents: $(w1_1 * w2_1) \times (w1_2 * w2_2)$ [cf. Widdows, 2008]. The quantitative interpretation of this metric corresponds to that of a scalar product, i.e., the higher the similarity score, the more related the components.

Thus, our task is to compare the actual vector of a multiword unit with that of the "composed" vector of its constituents, whereas the "composed vector" is defined by four models of compositionality described above. Figure 7.2 exemplifies the idea behind the overall procedure.

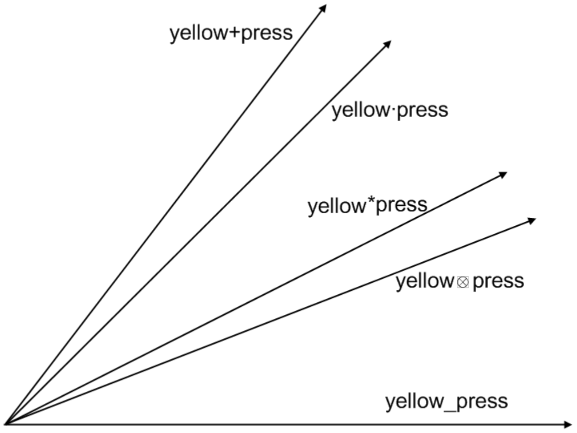

Figure 7.2: Composition operations on MWU in a 2-dimensional word space

7.1.2 Experimental Setup

In this work we make use of the Word Space Model (WSM) [Schütze, 1993] where the meaning of a word is modelled as an n-dimensional vector with dimensions being its co-occurrence signature derived via Random Indexing [Sahlgren, 2005].

We build our WSM on the excerpt of a local German newspaper corpus[1]. As our MWU test set we use a database of German (Preposition)-Noun-Verb (PNV) pairs available as an example data collection in the UCS-Toolkit[2]. From this database only word combinations with frequency of occurrence more than 30 in the corpus were considered.

The Semantic Vectors package [Widdows and Ferraro, 2008] was used to build the context vectors of reduced dimensionality. We use a context window of 15 words and limit the dimensionality to 100, resulting in 100 dimensional "meaning" - vectors for each word. In our experiments, MWUs as a whole also got

[1] Süddeutsche Zeitung (SZ) corpus for 2003 with about 42 million words.

[2] www.collocations.de.

assigned such meaning vectors using the same procedure. The meaning vectors for component words were always computed from contexts in which they appear alone, that is, not in the local context of the other constituent, in order to exclude the biasing contribution of the latter. Table 7.1 illustrates a possible resulting matrix, which indicates that, e.g., the words *gossip* and *celebrity* occur 20 times with *yellow_press* within a distance of 7 words before or 7 words after the *yellow press*.

	dim1=gossip	**dim3=celebrity**	**dim4=banana**	...	**dim100=resources**
yellow	0	0	20	...	0
press	1	3	0	...	15
yellow_press	20	20	0	...	1

Table 7.1: An example of a word space containing MWU

To evaluate the method, we use the manually annotated collocations database described by Evert and Krenn [2001] as our gold standard. This collection includes collocations that have been manually classified into Support Verb Constructions(SVC), figurative expressions, or neither of the two. SVC are (preposition-)noun-verb constructions where a noun provides the main semantic contribution to the meaning of the whole phrase, like in "Peter *took a walk*", or an example in German could be "Peter hat das Problem *in Angriff genommen*". The whole word combination in the case of SVC is neither non-compositional nor can it be called compositional. To be on the safe side, the current evaluation is based solely on the phrases annotated as figurative, as they are per se non-compositional. The latter constitute 19% of our test set (19 out of 100).

The idea behind our evaluation strategy is to use these non-compositional collocations to compare how reliable different vector composition models can identify them. This should give us a clue whether using a more advanced mathematical operator could be good enough to reproduce semantic composition in language.

7.1.3 Results

The resulting vector similarity values for tensor product range from -0.009 to 0.55; for vector sum, the cosine values are between 0.04 and 0.79; the products range from -0.009 to 0.03; and finally, convolution ranges from -0.04 to 0.66[1]. Since we cannot directly compare the values between the different composition operations, important are the comparisons within the individual models.

In computational linguistics, one straightforward way of doing this is by means of precision and recall.

Precision is defined in this case as the proportion of *true* multiword units under the given cosine similarity threshold. Recall is the proportion of multiword units under the cosine threshold out of all given MWUs. As there is generally a trade-off between the two measures, the F-measure [Manning and Schütze, 1999] is often used instead, which is a weighted harmonic mean of precision and recall. Table 7.2 gives an overview of precision, recall and F-score values for different cut-offs of a similarity value for the evaluated composition models.

ADDITION	< 0.2	< 0.3	< 0.4	< 0.5
Precision	0.125	0.28	0.29	0.25
Recall	0.05	0.53	0.84	0.88
F-measure	0.09	0.37	0.43	0.40
MULTIPLICATION	< 0.001	< 0.01	< 0.02	< 0.03
Precision	0.19	0.20	0.19	0.19
Recall	0.47	0.79	0.89	1.00
F-measure	0.27	0.39	0.31	0.31
TENSOR	< 0.03	< 0.05	< 0.1	< 0.15
Precision	0.21	0.29	0.31	0.28
Recall	0.16	0.37	0.84	1.00
F-measure	0.18	0.325	0.45	0.44
CONVOLUTION	< 0.01	< 0.1	< 0.2	< 0.26
Precision	0.22	0.20	0.22	0.25
Recall	0.26	0.47	0.79	1.00
F-measure	0.24	0.28	0.35	0.40

Table 7.2: MWU dataset: similarity values for evaluated compositionality models

The precision - recall diagram (Figure 7.3) demonstrates that tensor product does a consistently better job at recognizing non-compositional multiword units.

[1] The biggest possible value is 1.0. Remember, the higher the similarity score, the more related the components.

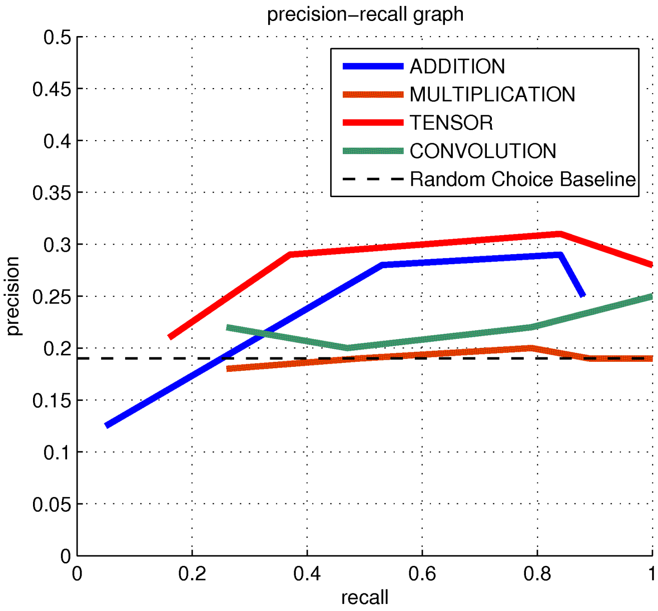

Figure 7.3: Precision-recall diagram

Though the precision of all the models seems to be rather small at first sight, it is worth mentioning that it is still significantly better than expected by chance alone for almost all models. The outcomes of other methods are rather dispersed in the vector space, especially those of vector addition. Our results are in line with those of Widdows [2008] who showed only on a couple of examples the predominance of tensor product.

Our findings show, on one side, that using a more advanced compositional operator, like tensor product, can lead to better results than vector addition, which is still the most common operator for vector composition. On the other side, it is obvious that just using a different mathematical operator with the same single vector-based models of word meaning representation is not sufficient. The latter motivated us to put in question the existing word space model paradigm in its current matrix-based form in respect to its ability to represent word meaning, and it pushed us to third-order tensors.

7.2 Graded Compositionality

Though the task of classifying phrases into compositional and non-compositional has long roots in computational linguistics, the problem that a binary classification may not be sufficient in many cases and that compositionality comes in degrees has been astonishingly ignored [Bannard et al., 2003].

Katz and Giesbrecht [2006] suggest that the technique for identifying non - compositional phrases by using vector space models provides a means, if rather a blunt one, for quantifying the degreee of compositonality of an expression.

The shared task, organized at the *Distributional Semantics and Compositionality* (DiSCo[1]) workshop, collocated with ACL-2011[2], addressed exactly this problem. The workshop attracted researchers interested in extracting non-compositional phrases from large corpora by applying distributional models that assign a graded compositionality score to a phrase as well as researchers interested in expressing compositional meaning with semantics space models.

Such a compositionality score is meant to denote the extent to which the compositionality assumption holds for a given expression. The latter can be used, for example, to decide whether the phrase should be treated as an idiom in the applications.

It is often the case that compositionality of a phrase depends on the context. Though we used the sentence context in the process of construction of the DiSCo gold standard, it was decided not to provide it with the dataset; thus, *a single compositionality score per phrase* was requested from the participating systems.

To the best of our knowledge, this task had not been addressed in the community until DiSCo-2011. It was the first attempt to offer a dataset and a shared task that allows to explicitly evaluate the models of graded compositionality.

[1] http://disco2011.fzi.de

[2] http://www.acl2011.org

7.2.1 Shared Task Dataset Description

For the shared task, we collected frequent phrases from the freely available WaCky[1] web corpora [Ferraresi et al., 2008]. They were already automatically sentence-split, tokenized, part-of-speech tagged and lemmatized, which reduced the load on both organizers and participants that decide to make use of these corpora.

We restricted candidate phrases to certain grammatical constructions to make the task more tangible.

Specifically, we use word pairs in the following relations:

- ADJ_NN: adjective modifying a noun, e.g. "red herring" or "blue skies"
- V_SUBJ: noun in subject position and verb, e..g. "flies fly" or "people transfer"
- V_OBJ: noun in object position and verb, e.g. "lose keys", "play song".

The target phrases were extracted semi-automatically from corpora by using part-of-speech patterns and sorting by frequency. Then the selected multiword units were manually evaluated for their validity as well as balanced for the presence of non-compositional phrases, as the latter are less spread than typical compositional phrases. If the candidates were completely randomly selected, an overwhelming number of compositional phrases would have biased the task for high compositionality.

After a candidate list was compiled, five sentences per candidate phrase were randomly selected and then manually filtered from the corpus. Figure 7.4 shows the sentences for "V_OBJ: buck trend" as an example output of this procedure.

The resulting dataset was manually annotated for compositionality with the help of crowdsourcing.

Amazon Mechanical Turk[2] is the pioneer of the crowdsourcing trend and offers a great possibility to construct manually annotated datasets quickly and at low cost in spite of certain disadvantages of crowdsourcing compared to manual annotations done by professional annotators. The latter include the possibly lower

[1] http://wacky.sslmit.unibo.it

[2] https://www.mturk.com/

– I would like to **buck** the **trend** of complaint !

– One company that is **bucking** the **trend** is Flowcrete Group plc located in Sandbach , Cheshire .

– " We are now moving into a new phase where we are hoping to **buck** the **trend** .

– With a claimed 11,000 customers and what look like aggressive growth plans , including recent acquisitions of Infinium Software , Interbiz and earlier also Max international , the firm does seem to be **bucking** the **trend** of difficult times .

– Every time we get a new PocketPC in to Pocket-Lint tower , it seems to offer more features for less money and the HP iPaq 4150 is n't about to **buck** the **trend** .

Figure 7.4: Sample of the data for **V_OBJ: buck trend** [Biemann and Giesbrecht, 2011]

quality of annotations, the non-comparability of annotators[1] and the rather high complexity of communicating complex tasks.

Using the experience of previous work [Biemann and Nygaard, 2010], the quality of annotations was guaranteed by a two-phase procedure. In the first step, a small data sample was annotated by a large number of so-called "workers"[2], who were asked to provide reasons for their decisions. Based on these explanations and the performance quality on this small dataset, a number of "workers" were chosen for the "real" final dataset, used in the shared task. In the second step, the selected annotators judged the phrases for their compositionality. Every phrase was annotated by 4 Amazon "workers", and every "worker" gave a score of 0 to 10 to every presented sentence. The scores of all annotators were averaged per phrase and normalized to the range of 0 to 100.

In the end, every phrase in the dataset received a compositionality score between 0 and 100, e.g., an adjective-noun phrase *little girl* has got a score of 93, a verb-object phrase *raise bar* - 9 and an adjective-noun word combination *second hand* - 14.

As too fine-scaled classification may indeed be "frustratingly hard"[Johannsen et al., 2011] and such a granularity may not be needed for the majority of practical applications, we provided a mapping of scores to three classes of compositionality, differentiating between low (0-25), medium (38-62) and high (scores of 75-100)

[1] It is rather tricky to compute inter-annotator agreement on such data.

[2] "Workers" are people accomplishing tasks on Amazon Mechanical Turk.

compositionality. We deliberately eliminated the ”borderline” cases to make the distinction more clear-cut. In contrast to the precise *numerical scores*, we define these scores as *”coarse”*.

7.2.2 Dataset Statistics

Datasets for English and German were constructed for the shared task. Only one participant submitted results for German.

Following most of the workshop participants and in order to provide comparable results, we concentrate on the English data for our current evaluation and report the numbers only for English[1]. Table 7.3 summarizes the English dataset quantitatively: with numbers in the cells denoting the amount of phrases in the *numerical* dataset and in brackets for the *coarse* set.

Per relation, the data was randomly split in approximatively 40% training, 10% validation and 50% test.

EN	ADJ_NN	V_SUBJ	V_OBJ	all
Train	58 (43)	30 (23)	52 (41)	140 (107)
Validation	10 (7)	9 (6)	16 (13)	35 (26)
Test	77 (52)	35 (26)	62 (40)	174 (118)
All	145 (102)	74 (55)	130 (94)	349 (251)

Table 7.3: DiSCo English dataset: number of target phrases (with coarse scores) [Biemann and Giesbrecht, 2011]

7.2.3 Evaluation Measures

Two official evaluation measures that were used for measuring the performance of the systems include the following:

1. **for numerical scores**: a score difference between the gold standard and a system’s response scores for corresponding target phrases;

[1] However, our approach is absolutely language independent and in the future we plan to evaluate it for further languages. Currently, it is out of the scope of this thesis.

For system's scores $S = \{s_{target1}, s_{target2}, ...s_{targetN}\}$ and gold standard scores $G = \{g_{target1}, g_{target2}, ...g_{targetN}\}$:

$$NUMSCORE(S,G) = \frac{1}{N} \sum_{i=1..N} |g_i - s_i|$$

Missing values in the system scores got assigned a default value of 50. A perfect score in this setup would be 0; indicating that there was no difference between system responses and the gold standard.

2. **for coarse scoring**: precision of coarse label predictions was used;

$$COARSE(S,G) = \frac{1}{N} \sum_{i=1..N} \begin{cases} s_i == g_i : 1 \\ otherwise : 0 \end{cases}$$

As with numerical scoring, missing system responses received a default value, in this case 'medium'. A perfect score would be 1.00, meaning a complete congruence of gold standard and system response labels.

3. additionally, two correlation values were provided for the evaluation of the participants score at the workshop: **Spearman's** *rho* and **Kendall's** *tau*. As in the end they are more or less linearly connected, we use just Spearman's *rho* for the detailed evaluation of parameters of DTSM in this section.

7.2.4 Participants of DiSCo and Official Results

Seven teams participated in the shared task. Table 7.4 summarizes the participants and their systems. Four of the teams - the University of Minnesota (Duluth), the University of York (UoY), the Jadavpur University (JUCSE) and the Trinity College Dublin (SCSS-TCD) - submitted three runs for the whole English test set. The team from Gavagai participated with two systems, one of which was for the entire English dataset (submission-ws) and another one included entries only for English V_SUBJ and V_OBJ relations (submission-pmi). The team from the UNED provided scores solely for English ADJ_NN pairs. The team from University of Copenhagen (UCPH) was the only one that delivered results for both English and German.

Systems	Institution	Team	Approach
Duluth-1 Duluth-2 Duluth-3	Dept. of Computer Science, University of Minnesota	Ted Pedersen	statistical association measures: t-score and pmi
JUCSE-1 JUCSE-2 JUCSE-3	Jadavpur University	Tanmoy Chakraborty, Santanu Pal, Tapabrata Mondal, Tanik Saikh, Sivaju Bandyopadhyay	mix of statistical association measures
SCSS-TCD:conf1 SCSS-TCD:conf2 SCSS-TCD:conf3	SCSS, Trinity College Dublin	Alfredo Maldonado-Guerra, Martin Emms	unsupervised WSM, cosine similarity
submission-ws submission-pmi	Gavagai	Hillevi Hägglöf, Lisa Tengstrand	random indexing association measure
UCPH-simple.en	University of Copenhagen	Anders Johannsen, Hector Martinez, Christian Rishøj, Anders Søgaard	support vector regression with COALS-based endocentricity features
UoY: Exm UoY: Exm-Best UoY: Pro-Best	University of York; Lexical Computing Ltd.	Siva Reddy, Diana McCarthy, Suresh Manandhar, Spandana Gella	exemplar-based WSMs prototype-based WSM
UNED-1: NN UNED-2: NN UNED-3: NN	NLP Group at UNED	Guillermo Garrido, Anselmo Peñas	syntactic VSM, dependency-parsed corpus, SVM classifier

Table 7.4: Participants of DiSCo-2011 Shared Task [Biemann and Giesbrecht, 2011]

Systems can be split into approaches based on statistical association measures and approaches based on word space models. On top, some systems used a machine-learned classifier to predict numerical scores or coarse labels.

The results of the official evaluation for English are shown in Tables 7.5 and 7.6. Table 7.5 reports the results for numerical scoring. *UCPH-simple.en* performed best with the score of 16.19. The second best system *UoY: Exm-Best* achieved 16.51, and the third was *UoY:Pro-Best* with 16.79.

The outcome of evaluation for coarse scores is displayed in Table 7.6. Here, *Duluth-1* performs highest with 0.585, followed closely by *UoY:ExmBest with 0.576* and *UoY: ProBest* with 0.567. *Duluth-1* is an approach purely based on association measures.

Both tables also report ZERO-response and RANDOM-response baselines. ZERO-response means that, if no score is reported for a phrase, it gets a default value of 50 (fifty) points in numerical evaluation and 'medium' in coarse evaluation. Random baselines were created by using random labels from a uniform distribution. Most systems beat the RANDOM-response baseline, only about half of the systems are better than ZERO-response.

Apart from the officially announced scoring methods, we provide Spearman's rho and Kendall's tau rank correlations for numerical scoring. Rank correlation scores that are not significant are noted in parentheses. With correlations, the higher the score, the better is the system's ability to order the phrases according to their compositionality scores. Here, systems *UoY: Exm-Best*, *UoY: Pro-Best / JUCSE-1* and *JUCSE-2* achieved the first, second and third best results respectively.

Overall, there was no clear winner for the English dataset. However, across different scoring mechanisms, *UoY: Exm-Best* was the most robust of the systems. The *UCPH-simple.en* system had the best performance in numerical evaluation overall and a very good performance on V_OBJ pairs but apparently uses a sub-optimal way of assigning coarse labels. The *Duluth-1* system, on the other hand, is not able to produce a numerical ranking that is significant according to the correlation measures, but does the best in the coarse scoring.

numerical scores	responses	ρ	τ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases				174	77	35	62
0-response baseline	0	-	-	23.42	24.67	17.03	25.47
random baseline	174	(0.02)	(0.02)	32.82	34.57	29.83	32.34
UCPH-simple.en	174	0.27	0.18	**16.19**	14.93	21.64	**14.66**
UoY: Exm-Best	169	**0.35**	**0.24**	16.51	15.19	**15.72**	18.6
UoY: Pro-Best	169	0.33	0.23	16.79	**14.62**	18.89	18.31
UoY: Exm	169	0.26	0.18	17.28	15.82	18.18	18.6
SCSS-TCD: conf1	174	0.27	0.19	17.95	18.56	20.8	15.58
SCSS-TCD: conf2	174	0.28	0.19	18.35	19.62	20.2	15.73
Duluth-1	174	(-0.01)	(-0.01)	21.22	19.35	26.71	20.45
JUCSE-1	174	0.33	0.23	22.67	25.32	17.71	22.16
JUCSE-2	174	0.32	0.22	22.94	25.69	17.51	22.6
SCSS-TCD: conf3	174	0.18	0.12	25.59	24.16	32.04	23.73
JUCSE-3	174	(-0.04)	(-0.03)	25.75	30.03	26.91	19.77
Duluth-2	174	(-0.06)	(-0.04)	27.93	37.45	17.74	21.85
Duluth-3	174	(-0.08)	(-0.05)	33.04	44.04	17.6	28.09
submission-ws	173	0.24	0.16	44.27	37.24	50.06	49.72
submission-pmi	96	-	-	-	-	52.13	50.46
UNED-1: NN	77	-	-	-	17.02	-	-
UNED-2: NN	77	-	-	-	17.18	-	-
UNED-3: NN	77	-	-	-	17.29	-	-

Table 7.5: Numerical evaluation scores for English @ DiSCo: average point difference and correlation measures; non-significant values are in parentheses

coarse values	responses	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases		118	52	26	40
zero-response baseline	0	0.356	0.288	0.654	0.250
random baseline	118	0.297	0.288	0.308	0.300
Duluth-1	118	**0.585**	0.654	0.385	0.625
UoY: Exm-Best	114	0.576	0.692	0.500	0.475
UoY: Pro-Best	114	0.567	**0.731**	0.346	0.500
UoY: Exm	114	0.542	0.692	0.346	0.475
SCSS-TCD: conf2	118	0.542	0.635	0.192	**0.650**
SCSS-TCD: conf1	118	0.534	0.64	0.192	0.625
JUCSE-3	118	0.475	0.442	0.346	0.600
JUCSE-2	118	0.458	0.481	0.462	0.425
SCSS-TCD: conf3	118	0.449	0.404	0.423	0.525
JUCSE-1	118	0.441	0.442	0.462	0.425
submission-ws	117	0.373	0.346	0.269	0.475
UCPH-simple.en	118	0.356	0.346	0.500	0.275
Duluth-2	118	0.322	0.173	0.346	0.500
Duluth-3	118	0.322	0.135	**0.577**	0.400
submission-pmi	-	-	-	0.346	0.550
UNED-1-NN	52	-	0.289	-	-
UNED-2-NN	52	-	0.404	-	-
UNED-3-NN	52	-	0.327	-	-

Table 7.6: Coarse evaluation scores for submitted results @ DiSCo

7.2.5 DTSM Evaluation with DiSCo Dataset

We evaluate the Distributional Tensor Space Model model on the DiSCo dataset and describe the procedure, the tested parameters and the detailed results in this section. After a thorough evaluation of the model parameters, we compare the achieved results of the DTSM to the DiSCo workshop participating systems.

7.2.5.1 Procedure

In order to build and evaluate the model, we proceed in the following way:

1. we build several DTSM models from the **BNC** corpus, which is much smaller than the ukWaC corpus that was used by the workshop participants (cf. Section 5.3);

   The models include all phrases, represented as single units, from the DiSCo dataset that were found in the BNC.

2. extract word matrices for component words as well as matrices for the complete phrases as a whole, using two ways of tensor interpretation - only the middle matrix or all three axes;

   For example, for the word pair *"ref herring"* we extract in the first case middle $Y-$ matrices for *"red"* and *herring*, as well as the $Y-$ matrix for the phrase as a whole *"red herring"*.

   In the second case, we would extract the $X-$ and $Y-$ slices for *"red"* and *herring* correspondingly and only the $Y-$ slice for *"red herring"*.

3. compute addition and multiplication for component word matrices obtained in the previous step, e.g.,

   ```
   addComposition(red herring)  = matrix (red) + matrix (herring)
   multComposition(red herring) = matrix (red) x matrix (herring)
   ```

4. compare the latter by means of cosine to the matrix of the expression used as a whole;

   ```
   add  = cosine(addComposition(red herring),matrix(red herring))
   mult = cosine(multComposition(red herring),matrix(red herring))
   ```

5. additionally to the numerical normalized score, we map the scores for the coarse values as well as calculate Spearman's ρ correlation for numerical scoring.

7.2.5.2 Parameters of the Model

The following modifiable parameters of Distributional Tensor Space Model have been thoroughly evaluated in this section:

1. choice of vocabulary for context dimensions;
2. number of context dimensions (dim);
3. context window size, or number of neighbours per side (nn);
   $context\_window\_size := nn \times 2 + 1$
4. number of decomposition factors (f);
5. weighting function ($boolean, frequency, pmi$);
6. compositional functions: addition (add) or multiplication ($mult$);
7. minimum number of triple occurrences (min);
8. tensor manipulation: using $\langle X, Y, Z\rangle$ slices instead of only Y ($\langle X, Y, Z\rangle$);

The numerical scores are computed straightforward within DTSM framework; for coarse scores we decide to map the resulting values between 0-32 to ”low”, 33-65 to ”medium” and 66-99 to ”high” in these preliminary parameter tuning experiments.

We successively test the influence of these parameters on the DiSCo dataset by means of fixing all the parameters except for the one that is under evaluation. For the sake of compactness, we will use the above enumerated abbreviations for the correspondingly fixed parameters in most cases.

By means of example, we will use a formulation like DTSM with $dim = 2000$, $min = 5$, $nn = 2$, $f = 50$ and varying compositional functions; which means that we build Distributional Tensor Space Model for the given experiment with 2000 context dimensions; minimum triple occurrence of 5 in the corpus; number of neighbours per side being 2 - that results in the context window size of 5; 50

decomposition factors and test it with different compositional functions (*add* and *mult*).

7.2.5.3 Results

In this following we present the results of the evaluation of DTSM with parameters listed in the previous section on the DiSCo shared task.

Choice of vocabulary for context dimensions. First, we evaluate the choice of words for context dimensions. We use DTSM with $dim = 5000$, $min = 5$, $nn = 6$ weighting function pmi and $f = 100$ for the test.

The context words were defined in the following 4 ways:

(I) all words in the context window that occur in the corpus more than 5 but less than 50 times except for stop words were used; OGDEN's[1] Basic English vocabulary list was added to the extracted list;

(II) the context word list is gained as in (I) but we restrict it to adjectives, adverbs, common and proper nouns;

For this, part of speech information, that is available with the BNC corpus, is used.

(III) same as [II] but with verbs and prepositions from the corpus;

(IV) only OGDEN Basic English vocabulary is used for context dimension.

numerical scores	responses	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
I	167	0.17	21.63	22.30	**21.88**	20.67
II	167	0.17	21.69	22.39	22.48	20.40
III	167	**0.19**	21.08	20.91	22.61	20.45
IV	55	0.11	**20.56**	**20.82**	24.33	**18.92**

Table 7.7: DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, pmi, $f = 100$ and *add* for different choices of context words

Table 7.7 reveals that using some linguistic preprocessing, like part of speech annotations, may overall slightly improve the performance of DTSM in its current

[1] http://ogden.basic-english.org/

form; however, not in all cases. Thus, using the basic vocabulary of simple English OGDEN **(IV)** without any part of speech information achieves the best numerical results in all categories except $subject - verb$ phrases. Obviously, using just basic school vocabulary for "English as a second language" of 850 words is enough to achieve very good performance for *adjective* – *noun* and $verb - object$ constructions, but not good enough for $subject - verb$ combinations. Adding to OGDEN **(IV)** the words of middle range frequency occurring 5 - 50 times in the corpus **(I)** drastically improves the result for $subject - verb$ phrases but decreases for the other grammatical types.

It is important to mention that we measure here the performance for the phrases that exist in the model and ignore the non-existing ones, as our interest is in the quality of the model and not in the winning of this special competition. Using only OGDEN vocabulary, only 55 phrases have got a compositionality score; the rest was not existing in the model. Evaluating OGDEN following the strategy that was used for the DiSCo workshop, i.e. by assigning missing values the score of 50, we receive the best correlation out of four tested setups for this parameter ($\rho = \mathbf{0.23}$ versus **0.11** for only found phrases, cf. Table 7.7).

To sum up, we observe that:

1. using certain context word vocabulary (here: OGDEN) may significantly improve the results for *adjective* – *noun* and $verb - object$ phrases but deteriorate the performance on $subject - verb$ constructions;
2. part of speech filtering enhances the success for *adjective* – *noun* units, decreases for $subject - verb$ combinations and doesn't seem to play a big role for $verb - object$ phrases; especially $subject - verb$ constructions are following different rules than the other grammatical types;
3. usage of possible all contexts without filtering is better for $subject - verb$ expressions.

Choice of vocabulary for context dimensions has surely immense impact on the model. However, this point needs further deeper explorations that are out of the scope of the current work.

Number of context dimensions (*dim*) and context window size (*nn*). Tables 7.8 and 7.9 show the results for two tested values for context dimensions ($dim = 2000$ and $dim = 5000$) as well as with three variations of context window size, i.e. with 1, 3 and 6 neighbours per side, resulting in context windows of size 3, 7 and 13.

We consider 2000 dimensions in analogy to the most successful matrix - based approaches [e.g. Mitchell and Lapata, 2010]. 5000 dimensions is an arbitrary chosen size for a bigger context.

We observe that the performance of the model gets better with more context dimensions and larger context windows; with a small deviation for $verb - object$ pairs for numerical scoring only.

Context window size is even more important than the number of context dimensions. The explanation could be that more "context" words get a chance to be considered with a larger context window.

numerical scores	responses	ρ	τ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases				174	77	35	62
2000, $nn = 1$	165	0.04	0.03	29.35	30.32	26.89	29.25
5000, $nn = 1$	165	0.04	0.03	29.04	30.14	26.68	29.00
2000, $nn = 3$	165	0.14	0.10	25.02	25.46	23.87	25.13
5000, $nn = 3$	165	0.13	0.09	24.76	25.14	23.34	25.08
2000, $nn = 6$	168	0.13	0.09	21.77	22.05	22.91	**20.77**
5000, $nn = 6$	168	**0.18**	**0.13**	**21.41**	**20.59**	**21.65**	22.29

Table 7.8: DiSCo dataset: average point difference and correlation measures for DTSM with $min = 5$, pmi, $f = 100$ and add for varying context dimensions and context window sizes

coarse values	responses	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases		118	52	26	40
2000, $nn = 1$	111	0.212	0.140	0.300	0.250
5000, $nn = 1$	111	0.271	0.211	0.269	0.350
2000, $nn = 3$	111	0.457	0.538	0.269	0.475
5000, $nn = 3$	111	0.483	0.577	0.308	0.475
2000, $nn = 6$	113	0.517	0.596	0.308	**0.550**
5000, $nn = 6$	113	**0.534**	**0.615**	**0.346**	**0.550**

Table 7.9: DiSCo dataset: coarse evaluation scores for DTSM with $min = 5$, pmi, $f = 100$ and add for varying context dimensions and context windows

Table 7.10 shows a similar tendency for matrix multiplication, i.e., the bigger the context window, the better the performance.

numerical scores	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
$dim = 2000$, $nn = 5$	0.11	25.20	27.38	20.61	25.05
$dim = 2000$, $nn = 13$	0.14	**21.73**	**23.16**	**17.88**	**22.05**

Table 7.10: DiSCo dataset: average point difference and correlation measures for DTSM with $min = 5$, pmi, $f = 100$, $mult$ and varying context window sizes

Number of decomposition factors. We further test the influence of the number of factors ($f = 50/100/200$) used for tensor decomposition as well the quality of the model without using dimensionality reduction.

numerical scores	ρ	τ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
no decomposition	0.13	0.10	62.35	64.17	55.94	63.72
50 factors	**0.21**	**0.14**	**20.74**	**20.15**	24.34	**19.43**
100 factors	0.18	0.13	21.41	20.59	21.65	22.29
200 factors	0.15	0.10	23.49	21.91	**20.94**	26.90

Table 7.11: DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, pmi, add and $f = 50/100/200$ or without decomposition

It is obvious from Tables 7.11 and 7.12 that tensor decomposition brings about significant improvement in performance compared to using no decomposition, when matrix addition as compositionality operation is used. However, more decomposition factors are not per se better.The best result was obtained for 5000 dimensions with only 50 factors; however, there seem to be differences in the influence of factor number on different types of grammatical relations: the less factors, the better for $adjective - noun$ and $verb - object$ phrases; with the opposite tendency for $subject - verb$ phrases.

Furthermore, Table 7.13 shows that using less factors brings about better performance for all kinds of grammatical constructions when using matrix multiplication. Decomposition is important also with *matrix multiplication* as otherwise too few results are found (55 out of 174) and the numerical score difference is more than twice as big (55.25 versus 24.53).

coarse values	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
no decomposition	0.068	0.077	0.00	0.100
50 factors	**0.542**	**0.635**	0.308	**0.575**
100 factors	0.534	0.615	0.346	0.550
200 factors	0.415	0.423	**0.461**	0.375

Table 7.12: DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, *pmi*, *add* and $f = 50/100/200$ or w/t decomposition

numerical scores	responses	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
no decomposition	55	(0.003)	55.25	55.14	52.00	56.58
50 factors	163	**0.10**	**24.53**	**27.46**	**16.67**	**25.45**
100 factors	163	0.08	28.37	29.84	21.36	30.52

Table 7.13: DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 10$, $nn = 13$, *pmi*, *mult* and $f = 50/100$ or without decomposition

Weighting functions. We evaluate here three weighting functions that are often used with word space models: boolean, simple frequency count and association measure *pointwise mutual information* (PMI).

numerical scores	responses	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases			174	77	35	62
boolean	168	(-0.004)	23.99	21.92	26.43	25.18
frequency	168	(-0.08)	24.43	22.00	27.03	25.65
PMI	168	**0.18**	**21.41**	**20.59**	**21.65**	**22.29**

Table 7.14: DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, $f = 100$, *add* and using *boolean*, $frequency$ and *pmi* weighting functions

Tables 7.14 and 7.15 provide evidence that PMI is the best weighting function out of the three tested ones in numerical evaluation as well as in accuracy and correlation.

Compositional functions. In the next step, we evaluate matrix compositionality operations of *addition* and *multiplication* for varying sizes of context windows and context dimensions.

Tables 7.16 and 7.17 reveal that for a bigger context window ($nn = 6$) *matrix multiplication* is very close to *matrix addition* in correlation values ($rho = 0.16$

coarse values	responses	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
boolean	113	0.474	0.557	0.307	0.475
frequency	113	0.459	0.564	0.304	0.425
PMI	113	**0.534**	**0.615**	**0.346**	**0.550**

Table 7.15: DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $nn = 6$, $f = 100$, *add* and using *boolean*, *frequency* and *pmi* weighting functions

numerical scores	responses	ρ	τ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases				174	77	35	62
mult, $nn = 1$	165	0.05	0.03	35.54	38.77	27.13	36.23
add, $nn = 1$	165	0.04	0.03	29.04	30.14	26.68	29.00
mult, $nn = 3$	165	0.05	0.03	30.65	32.52	22.83	32.74
add, $nn = 3$	165	0.13	0.09	24.76	25.14	23.34	25.08
mult, $nn = 6$	168	0.16	0.11	24.85	27.48	**18.80**	25.00
add, $nn = 6$	168	**0.18**	**0.13**	**21.41**	**20.59**	21.65	**22.29**

Table 7.16: DiSCo dataset: average point difference and correlation measures for DTSM with $dim = 5000$, $min = 5$, $f = 100$, *pmi* and *add* or *mult*

and $rho = 0.18$ correspondingly). However, multiplication is consistently better for $subject - verb$ constructions than addition, except for the smallest context window of 3 (with $nn = 1$); while addition performs better for $adjective - noun$ and $verb - object$ phrases.

coarse values	responses	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases		118	52	26	40
mult, $nn = 1$	111	0.237	0.192	0.423	0.175
add, $nn = 1$	111	0.271	0.211	0.269	0.350
mult, $nn = 3$	111	0.322	0.269	**0.538**	0.250
add, $nn = 3$	111	0.483	0.577	0.308	0.475
mult, $nn = 6$	113	0.381	0.327	0.461	0.400
add, $nn = 6$	113	**0.534**	**0.615**	0.346	**0.550**

Table 7.17: DiSCo dataset: coarse evaluation scores for DTSM with $dim = 5000$, $min = 5$, $f = 100$, *pmi* and *add* or *mult*

Number of triple occurrences A further parameter that we test is the minimum number of triple occurrences, i.e., the minimum number of times a triple has to occur in the corpus in order to be included into the model. Tables 7.18

and 7.19 demonstrate that using the minimum of 5 occurrences is better for correlation results than using 3 or 10. The numerical difference does not show any particular preference, except that the minimum of 10 is generally worse. However, we should treat this outcome with caution. It may be the case, that many contexts do not pass the bigger threshold for the triple occurrence with such a relatively small corpus (**BNC**); thereby causing worse performance. This parameter should be further evaluated with a bigger corpus, like **ukWaC** (cf. Section 5.3).

numerical scores	responses	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases			174	77	35	62
$min = 3$	174	0.11	20.20	20.23	24.72	**17.64**
$min = 5$	174	**0.18**	**19.68**	**19.57**	**21.12**	19.02
$min = 10$	167	0.08	22.11	21.05	25.45	21.58

Table 7.18: DiSCo dataset: average point difference and correlation measures for DTSM with *dim* = 2000, f = 100, *nn* = 13, *pmi*, *add* and using varying minimum triple occurrences size

numerical scores	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
2000, $min = 3$	0.07	**19.72**	**22.16**	18.15	**17.76**
2000, $min = 5$	**0.14**	21.73	23.16	**17.88**	22.05
2000, $min = 10$	0.05	28.29	30.43	20.03	30.18

Table 7.19: DiSCo dataset: average point difference and correlation measures for DTSM with *dim* = 2000, f = 100, *nn* = 13, *pmi*, *mult* and using varying minimum triple occurrences size

Tensor Manipulation. Last but not least, we examine $3d$ - structure of the tensor by extracting not only the "middle" Y slice (cf. Figure 5.2) matrix but also the "left" X slice (cf. Figure 5.1) and the "right" Z slice (cf. Figure 5.3) matrices for word representation in certain grammatical constructions.

To be precise:

- for *subject-verb* and *adjective-noun* phrases, we extract the X and Y slices to get *subject/adjective* and *verb/noun* matrices correspondingly;
- for *verb-object* phrases, we extract the Y and Z slice matrices.

$\langle X, Y, Z\rangle$ row in Table 7.20 shows the results for this way of tensor manipulation for DiSCo dataset with the so far best performing DTSM. We have evaluated this setting with a number of further parameter modifications, i.e. varying *dim*, *nn* and *f* parameters. However, they all show similar tendencies: the numerical score difference for all kinds of grammatical constructions is significantly worse in the $\langle X, Y, Z\rangle$ setting compared to using only the middle matrix. However, the Spearman's ρ coefficient is higher for $adjective - noun$ and $verb - subject$ combinations when using X and Y axes.

Furthermore, we report the score of the heuristics (”mixed”) based on the previously described evaluations. ”Mixed” means that *matrix multiplication* is used for *subject-verb* constructions and *matrix addition* for *adjective-noun* and *verb-object* units. ”Mixed” rule achieves the currently best correlation for DTSM without differentiating between grammar types.

	composition model	all (ρ)	ADJ_NN (ρ)	V_SUBJ (ρ)	V_OBJ (ρ)
$dim = 5000$	add	21.33 (0.15)	**20.23** (0.18)	27.03 (-0.09)	19.5 (**0.29**)
	mult	**19.04** (0.12)	21.25 (0.20)	**15.94** (0.11)	**18.12** (0.10)
	mixed	19.09 (**0.26**)			
$dim = 5000$	add	35.87 (0.05)	36.58 (**0.24**)	29.58 (**0.25**)	38.48 (-0.29)
$\langle X, Y, Z\rangle$	mult	46.84 (0.09)	50.69 (0.15)	44.18 (-0.29)	43.75 (0.14)
	mixed	38.80 (0.02)			

Table 7.20: DiSCo: average point difference and correlation measures using $\langle X, Y, Z\rangle$ slices with $dim = 5000$, $min = 5$, $nn = 13$, $f = 50$, *pmi* and different compositionality models (*add*, *mult*, *mixed*)

Qualitative Insight. We further evaluate the numerical score difference on the coarse dataset. As a quick reminder, the coarse dataset contains only those phrases that received scores between 0-25 (low), 38-62 (medium) and 75-100 (high) in the process of manual annotation. If we restrict the numerical evaluation to this smaller and more carefully chosen dataset, the system's performance gets significantly better, especially for correlation.

We show that on two example runs: with 2000 dimensions and with 5000 dimensions. Best overall numerical or correlation results are marked **bold** and simply best result within each of the runs is *italicized* in Table 7.21.

Apparently, the "borderline" cases of compositionality, that were eliminated for the coarse dataset, decrease the performance of the computational system substantially. Maybe, the first (numerical) dataset should have been cleaned from these phrases for "cleaner" evaluation.

Another observation from these two tests that is worth mentioning is that with smaller number of dimensions (here: $dim = 2000$) and smaller context window (here: $nn = 2$) *matrix multiplication* achieves better results for all kinds of grammatical constructions than *addition*. Vice versa, *matrix addition* works better with bigger number of dimensions ($dim = 5000$) and bigger context window ($nn = 6$) as well as given more occurrences in corpus ($min = 5$).

	min	nn	composition	all (ρ)	ADJ_NN (ρ)	V_SUBJ (ρ)	V_OBJ (ρ)
2000	2	2	add	22.15 (0.11)	20.16 (0.20)	29.69 (-0.09)	20.52 (0.39)
coarse			mult	**18.67** (-0.03)	*19.92* (0.12)	*17.39* (0.06)	**17.8** (-0.08)
			mixed	19.81 (0.33)			
2000	2	2	add	22.31 (0.12)	*24.125* (0.05)	21.88 (0.21)	*20.31* (0.32)
num			mult	27.18 (0.05)	33 (0.06)	**17.36** (**0.14**)	25.53 (0.10)
			mixed	*21.40* (0.17)			
5000	5	6	add	*20.72* (0.27)	**19.62** (**0.38**)	*21.4* (-0.26)	*21.76* (**0.42**)
coarse			mult	32.63 (0.19)	35.28 (0.26)	24.28 (-0.09)	34.70 (0.37)
			mixed	21.37 (**0.35**)			
5000	5	6	add	21.08 (0.19)	*20.91* (0.18)	22.61(-0.26)	*20.45*(0.34)
num			mult	24.40 (0.11)	26.76 (0.20)	*19.76* (0.04)	24.05 (0.11)
			mixed	*20.51* (0.25)			

Table 7.21: DiSCo: average point difference and correlation measures for the "coarse" dataset compared to the "numerical dataset"; with $f = 100$ and weighting function PMI

Obviously, more data and decomposition compensates somehow the obvious theoretical advantages of *multiplication* in such a model.

Table 7.22 lists a number of phrases from the dataset with their grammatical roles (column 1) and their *gold* scores (column 3). These are the phrases where the difference in numerical score assigned by the system and the gold standard score is bigger than 40. Phrases from **Group I** are *highly* compositional expressions that got small compositionality scores by DTSM and, vice versa, a relatively few low compositional phrases received high scores from the system (**Group II**).

High error rate for highly compositional phrases may be due to the fact, that these phrases as well as their components are so wide spread that their contexts are no more distinctive per se. Presumably, a more sophisticated strategy for context word choice is needed in this case.

gram. type	phrase	gold standard score
Group I		
EN_V_SUBJ	event occur	92
EN_V_SUBJ	error occur	85
EN_ADJ_NN	panoramic view	82
EN_V_SUBJ	child want	91
EN_ADJ_NN	broad range	85
EN_ADJ_NN	rechargeable battery	96
EN_V_SUBJ	evidence show	60
EN_V_OBJ	obtain information	93
EN_ADJ_NN	high mountain	92
EN_V_OBJ	help people	97
EN_ADJ_NN	civil war	80
EN_V_OBJ	provide training	90
EN_V_OBJ	develop methods	91
EN_V_OBJ	help children	90
EN_V_OBJ	find way	86
EN_V_OBJ	promote excellence	81
Group II		
EN_V_OBJ	lose sight	19
EN_V_OBJ	take plunge	15
EN_ADJ_NN	social capital	46
EN_ADJ_NN	red tape	11
EN_V_OBJ	foot bill	15
EN_V_OBJ	raise bar	9
EN_ADJ_NN	second hand	14

Table 7.22: DiSCo dataset: excerpt of phrases where numerical score difference between *gold standard* and *system* is bigger than 40

7.2.6 Summary

Table 7.23 summarizes our observations from the above results.

In order to avoid over-generalization, we use concrete numbers from the experiments; i.e. instead of claiming that more dimensions are generally better, we say that the model with 5000 dimensions performed better than the one with 2000 dimensions based on the evaluations described in this section. It does not necessarily mean, that a model with 10000 dimensions would be better than the one with 5000. Consequently, to say that more dimensions are always better may

	Multiplication	**Addition**
context size (*dim*)	not significant	$dim = 5000 > dim = 2000$
neighbours per side (*nn*)	$nn = 13 > nn = 5\|\|2$	$nn = 6 > nn = 1\|\|3$
decomposition factors (f)	decomposition > no decomposition $f = 50 > f = 100$ for all constructions	decomposition > no decomposition $f = 50$ better for ADJ_N and V_OBJ $f = 200$ better for V_SUBJ
weighting function	PMI	PMI
composition operation	better for V_SUBJ	better for ADJ_N and V_OBJ
usage of $\langle X, Y, Z \rangle$ slices	worse numerical results	worse numerical results

Table 7.23: Summary of DTSM Evaluation on DiSCo Dataset

turn out to be false. We use in some cases the sign " > " for *"better"* in the summary table.

We've attested with the above-described experimental setup that:

- the number of context dimensions is especially important when using matrix addition;
- the context window size matters for both operations: bigger window is better;
- decomposition brings about better correlation and better recall;
- smaller number of factors seems to affect matrix multiplication in a positive way; when using addition - less factors appears to be profitable for *adjective-noun* and *verb-object* while more factors are better for *subject-verb* phrases;
- matrix multiplication is consistently better for *subject-verb* constructions; while addition favours more *adjective-noun* and *verb-object* objects;
- tensor manipulation by using X and Z slices additionally to middle Y matrices didn't improve the model.

We leave the question of how many examples per target word or phrase is needed to train a model as well as the choice of context words open for future research.

Based on the above tests, we compare the performance of the so-far best DTSM model to the results of the DiSCo shared task participants. Our best model includes 5000 context dimensions, a context window of size 27, minimum 5 triple occurrences in corpus, 50 factors for decomposition and compositional operations

numerical scores	ρ	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
DTSM$_{add}$	0.15	21.33	20.23	27.03	19.5
DTSM$_{mult}$	0.12	19.04	21.25	15.94	18.12
DTSM$_{mixed}$	0.26	19.09			
0-response baseline	-	23.42	24.67	17.03	25.47
random baseline	(0.02)	32.82	34.57	29.83	32.34
UCPH-simple.en	0.27	**16.19**	14.93	21.64	**14.66**
UoY: Exm-Best	**0.35**	16.51	15.19	**15.72**	18.6
UoY: Pro-Best	0.33	16.79	**14.62**	18.89	18.31
UoY: Exm	0.26	17.28	15.82	18.18	18.6
SCSS-TCD: conf1	0.27	17.95	18.56	20.8	15.58
SCSS-TCD: conf2	0.28	18.35	19.62	20.2	15.73

Table 7.24: DTSM$_{best}$ (dim = 5000, min = 5, nn = 13, f = 50, pmi) compared to the best DiSCo participating systems: numerical scores and overall correlation

of addition for $ADJ - NN$ and $V - OBJ$ as well as simple multiplication for $V - SUBJ$ (see Table 7.24).

Based on the development set, we optimized the parameters for mapping from numerical to coarse scores: the values between 0-20 are mapped to ”low”, 21-77 to ”medium” and 78-100 to ”high” similarity groups (Table 7.25).

coarse values	responses	EN all	EN_ADJ_NN	EN_V_SUBJ	EN_V_OBJ
number of phrases		118	52	26	40
DTSM$_{best}$	113	**0.61**	0.58	**0.68**	0.61
zero-response baseline	0	0.356	0.288	0.654	0.250
random baseline	118	0.297	0.288	0.308	0.300
Duluth-1	118	0.585	0.654	0.385	0.625
UoY: Exm-Best	114	0.576	0.692	0.500	0.475
UoY: Pro-Best	114	0.567	**0.731**	0.346	0.500
UoY: Exm	114	0.542	0.692	0.346	0.475
SCSS-TCD: conf2	118	0.542	0.635	0.192	**0.650**

Table 7.25: DTSM$_{best}$ compared to DiSCo participating systems: coarse scores

Our system is better than both baselines, but achieves only the seventh place based on the overall numerical evaluation (Table 7.24). Still, DTSM is the second best for $subject - verb$ constructions and only insignificantly worse than the best DiSCo system for this kind of grammatical construction: 15.94 (**DTSM**) versus

15.72 (**UCPH-simple.en**). As a quick reminder, **UCHP** is a machine learning system, based on support vector regression. The **UoY** and **SCSS** systems are based on word space models. **Duluth** is based on statistical association measures, i.e. on "first-order" statistics of word co-occurrence.

Thus, DTSM achieves the second best results for $subject - verb$ phrases among word space models and comparable results for other constructions, taking into consideration that we've used a much smaller corpus for our experiments than the other participants (**BNC** with hundred million versus **ukWaC** with two billion words) and that the original dataset was based on the **ukWaC**. The latter may have offered a little bias for competing systems, apart from the fact that the size of the corpus is particularly important for distributional models. Another notable fact is that we've obtained the **best overall result** in the **coarse evaluation** setup (Table 7.25); mostly due to the brilliant scores for $subject - verb$ (the best) but also very good results for $verb - object$ (third best) phrases.

All-in-one, Distributional Tensor Space Model seems to offer a new perspective for sentential composition. We prove it here mostly by means of very good results for $subject - verb$ composition but also positive outcomes for $verb - object$ constructions. The former are not particularly explored so far, as the majority of research on compositionality has been concentrated on $adjective - noun$ or $noun - noun$ combinations.

7.3 Semantic (Non-)Compositionality in Context

SemEval-2013, Task 5b[1], was a further evaluation campaign co-organized by the author of this thesis. It was an extension of the ideas that emerged within the evaluation efforts described in the previous two sections.

The task was to let computational systems decide whether a target phrase is used in its literal or figurative meaning in a given context. For example, *"big picture* might be used **literally** as in *"Click here for a bigger picture"* or **figuratively** as in *"To solve this problem, you have to look at the bigger picture"*. Another example could be an expression *"carve in stone"* in *"reliefs carved in stone"*

[1] http://www.cs.york.ac.uk/semeval-2013/task5/

```
142847 carve in stone figuratively This is complete nonsense .
Marriage and the family are not institutions <b>carved</b> <b>in</b>
<b>stone</b> throughout the ages .  Their forms have undergone radical
change at various points in history.


4288 carve in stone literally On an exterior wall above the main
entrance.  Description :  3 square reliefs <b>carved</b> <b>in</b>
<b>stone</b>, each depicting a scene representing an aspect of
childhood and learning.
```

Figure 7.5: Example of SemEval, Task 5b, Dataset

versus *"Marriage and the family are not institutions carved in stone throughout the ages"*.

The dataset consists of real usage examples of such phrases from the **ukWaC** corpus, with target units marked by **html bold tag** $< b >$ (see Figure 7.5).

Each phrase in the dataset is offered in at most 5 different contexts. For each context, the task is to decide whether the target phrase is used literally or figuratively here. There are two subtasks within this task. Both of them are structured in a similar way: there is a *training*, *development* and *test* set. *Training set* is used for model construction or "training", depending on the method. *Development set* is usually used for parameter optimization of the "trained" model; and *test set* is used for model evaluation.

The difference is in the choice of phrases: while in the first subtask the same phrases are being used in the train, development and test sets, different phrases are used in the second subtask.

Therefore, there are two datasets:

1. "unknown phrases setting" (called by the organizes *"all words"* in analogy to machine learning terminology) - where one set of phrases with their literal and figurative uses is included in the *training set* and another set of phrases in the *development* and *test sets*;

2. "known phrases task" (defined as *"lexical sample"* in the task) - where the same phrases are in the training and test sets.

The task of classifying unknown phrases is more difficult per se, as the systems can not train models on known examples. These two tasks address, therefore, *supervised* and *unsupervised* systems at the same time.

7.3.1 Procedure

In the *"all words"* setup, we proceed in the same way as in the previous experiments. For each target phrase:

1. we collect 200 examples for each of the component words from the **ukWaC** corpus,
2. build the DTSM model from these example sentences, ignoring *stop words*,
3. extract word matrices for component words,
4. compute addition and multiplication for word matrices,
5. if the result is bigger than a certain threshold, defined on the training set, than a phrase is classified as *literal*; otherwise as *figurative*.

For the second dataset type (*"lexical sample"*) , we combine the training and development sets and use both for training of literal or figurative contexts, in the same way as Experiment I was conducted in Katz and Giesbrecht [2006]. We calculate the literal and idiomatic vectors for every multiword unit on the basis of the training and development data and calculate the cosine between a phrase in the target context and these two vectors. If the cosine of the target phrase vector with the literal vector is bigger, then the system classifies it as literal and vice versa.

The performance is measured by means of precision, which is in this case defined as follows:

$$\mathrm{P} = \frac{number\ of\ correctly\ classified\ phrases}{all\ phrases\ in\ the\ test\ set} \quad (7.3.1)$$

7.3.2 Results

We've done a qualitative check of the model by comparing the average score assigned by the model to *literal* and *figurative* vectors. As Table 7.26 testifies, our DTSM model consistently and correctly assigns bigger values for *literal* usages and smaller ones for *figurative*; with a more clear cut difference for *matrix multiplication.*

Another observation is that parameter modification, such as the number of dimensions (2000 versus 3000) and the size of the context window ($nn = 5/13$), does not seem to make a big difference for the *addition* operation; while it matters a lot for *multiplication.* The average values vary between $47 - 58$ for the *literal* usage and $46 - 56$ for the *figurative* one when using addition; for *multiplication*: $17 - 41$ and $13 - 37$ correspondingly.

parameters	add_{lit}	add_{fig}	$mult_{lit}$	$mult_{fig}$
$dim = 2000$, $nn = 5$, $f = 50$	58	56	41	37
$dim = 2000$, $nn = 13$, $f = 50$	56	53	30	25
$dim = 2000$, $nn = 13$, $f = 100$	48	46	18	13
$dim = 3000$, $nn = 13$, $f = 100$	51	50	24	19
$dim = 3000$, $nn = 13$, $f = 200$	47	46	17	13

Table 7.26: SemEval 2013 Task 5b: Average cosine similarity values for literal and figurative vectors

Similarly to the other tasks, we report here the precision for both experiments using different model parameters. Figure 7.27 displays the results for two sizes of context dimensions (2000 and 3000), two context windows (with number of neighbours per side 5 and 13) and three varying numbers for decomposition factors (50, 100 and 200).

Similarly to the results obtained with other datasets so far, we observe that for unattested contexts:

- for matrix addition: the results get consistently better with more dimensions, more decomposition factors and bigger context window;

# of dimensions	# of nn per side	factors	P_{all_words}		P_{lex_sample}
			ADD	MULT	
2000	5	50	0.55	0.45	0.58
2000	13	50	0.55	0.60	**0.63**
2000	13	100	0.59	0.42	0.56
3000	13	100	**0.62**	0.66	0.62
3000	13	200	**0.62**	**0.67**	0.60

Table 7.27: SemEval 2013 Task 5b: DTSMresults

– matrix multiplication is not linearly improving with bigger parameter sizes, but we manage to achieve the best results here with matrix multiplication, 3000 dimensions, max. 13 neighbours per side and 200 decomposition factors.

For ”known” contexts, we get the best precision with 2000 dimensions, max. 13 neighbours and 50 factors.

Table 7.29 shows the performance of our currently best model compared to the top systems of SemEval (Task 5b) contest.

Baseline MFC (”Most Frequent Class”), provided by the organizers, simply assigns the most frequent class; in this dataset it is ”figurative”.

UNAL [Jimenez et al., 2013], the ”winner” system for the unseen phrases is a machine learning approach where a logistic classifier, based on part-of-speech tags, stylistic features and distributional statistics, is used. Machine learning approaches are out of the scope of this work; so we do not go into further details here.

IIRG Run3 [Byrne et al., 2013] is the so-called *”word overlap”* method, where a simple bag-of-words is created for each target phrase. Only nouns that occurred more than twice in the context of the phrase were recorded and labelled as *figurative* or *literal*. This approach is *”distributional”* in its nature but it is based only on superficial first-order contexts. Therefore, **IIRG** didn’t submit any runs for the setting with ”unseen phrases” as it can be used only as long as the previously attested phrases as well as similar contexts are used.

The most successful run from **IIRG** that did the best in differentiating between *figurative* and *literal* usages includes only nouns as contexts. Therefore, we test

# of dimensions	# of nn per side	factors	P_{all_words}		P_{lex_sample}
			ADD	MULT	
2000, 5-50	13	100	0.64	**0.67**	0.59
2000, nouns	13	100	0.60	0.65	0.60
2000, all	13	100	0.59	0.42	0.56

Table 7.28: SemEval 2013 Task 5b: evaluating DTSM for different context choices

System	P_{all_words}		P_{lex_sample}
	ADD	MULT	
DTSM$_{best}$	0.64	**0.673**	0.63
Baseline MFC	0.503		0.616
SemEval Best System for known: IIRG	–		**0.779**
SemEval Best System for unseen: UNAL	0.668		0.754

Table 7.29: SemEval 2013 Task 5b: DTSM$_{best}$ versus other models

also this setup with DTSM, i.e. we restrict our context words to *nouns* except that we add words from the test set (see row *"2000 nouns"* in Table 7.28). Table 7.28 shows the precision of DTSM for three different context dimension strategies:

1. **5-50** - only words that occur between 5 and 50 times in the corpus are taken as context dimensions;
2. **nouns** - only nouns build the contexts;
3. **all** - all words that co-occur with target phrases are considered; except for stop words.

All variations of our model are better than the baseline. Though the task for classification of *unseen phrases* is more difficult, it is not the case for our system. Our approach seems to work even better for the *unknown contexts.* The **DTSM multiplicative model** using words that occur 5-50 times in the corpus as context words achieves **the best result** of all the systems in this setup.

The performance for the *known phrases* is worse than that of the participating systems. The latter is most probably due to a very small number of "training" contexts, at most 5 per *literal* and *figurative* usage which is too small for distributional models; but obviously enough for participating machine learning systems, mostly because they use lexicalised contextual clues.

7.4 Phrasal Semantics

Apart from self-constructed evaluation resources, we perform experiments on the datasets that have become more or less benchmarks for compositional distributional models.

The dataset[1] of Mitchell and Lapata [2008, 2010] contains pairs of adjective-noun, verb-object and compound noun phrases, and the task is to compare two phrases or simple sentences for their similarity. This evaluation setup was suggested by Kintsch [2001]. However, Kintsch himself demonstrated his algorithm only on a few selected examples; this way of evaluation was later criticized in literature [Frank et al., 2008]. Nevertheless, the idea suggested by Kintsch [2001] was taken up later and extended to larger and proper evaluation datasets.

In Mitchell and Lapata [2010], the participants were asked to rate similarity between each pair of phrases on a 1-7 scale. The following example shows that participant 1 gave a similarity score of 5 to the verb-object combination ”use knowledge” and ”exercise influence”, and a score of 1 to ”begin career” and ”suffer loss”:

```
participant1 verbobjects 2 use knowledge exercise influence 5
participant1 verbobjects 2 begin career suffer loss 1
```

	Adjective-Noun	Noun-Noun	Verb-Object
	Mean SD SE	Mean SD SE	Mean SD SE
High	3.76 1.926 0.093	4.13 1.761 0.085	3.91 2.031 0.098
Medium	2.50 1.814 0.087	3.04 1.732 0.083	2.85 1.775 0.085
Low	1.99 1.353 0.065	2.80 1.529 0.074	2.38 1.525 0.073

Table 7.30: Descriptive statistics for Mitchell and Lapata [2010] dataset: Human performance

The aim was to collect phrases that were representative for three coarse classes of similarity: high, medium and low. The high-similarity items were chosen from phrases occurring more than 100 times in the BNC. The reliability of collected

[1] The dataset can be obtained here: http://homepages.inf.ed.ac.uk/s0453356/share

items was evaluated by testing if difference in the subjects' ratings was significant ($p < .01$) and by measuring mean, standard deviation and standard error for the similarity ratings for items within each of the three groups: high, medium and low similarity. Table 7.30 shows that the mean ratings demonstrate at least the correct ordering of manual scores from high to low.

7.4.1 Procedure

For this exercise, we proceed in the similar way as in **graded compositionality** and **semantic (non-)compositionality** tasks described in the previous two sections:

1. build the DTSM model from the **BNC** corpus;
2. extract word matrices for component words as well as phrases, using two ways of tensor interpretation - only the middle matrix or all three axes;

   For example, for two word pairs - *"knowledge use"* and *"influence exercise"* - that the system should assign a similarity score to, we extract in the first experiment middle $Y-$ matrices of *"knowledge, use, influence and exercise"*, as well as middle matrices for multiword units (mwu) as a whole *"use knowledge"* and *"exercise influence"*.

   In the second experiment, for *noun1-noun2 (compound nouns)* and *adjective-noun* phrases, X and Y slices are extracted for $noun1$ or $adjective$ and $nouns2$ or $noun$ matrices correspondingly; for *verb-object* phrases, we extract the Y- and Z- slice matrices. This setting is defined as "xyz" in Tables 7.33 and 7.34.
3. compute addition and multiplication for component word matrices obtained in the previous step;

   ```
   addComposition1 = matrix (use) + matrix (knowledge)
   addComposition2 = matrix (exercise) + matrix (influence)

   mulComposition1 = matrix (use) x matrix (knowledge)
   mulComposition2 = matrix (exercise) x matrix (influence)
   ```
4. compare those by means of cosine;

```
score1 (add) = cosine (addComposition1, addComposition2)
score2 (mult) = cosine (mulComposition1, mulComposition2)
score3 (mwu) = cosine (mwu1, mwu2);
```

5. calculate Spearman's ρ correlation between the resulting three scores and the gold standard.

7.4.2 Results

Similarly to Mitchell and Lapata [2010], we measure first the reliability of our system's classification by measuring mean, standard deviation and standard error for the similarity ratings for items within each of the three groups: high, medium and low similarity on the example of DTSM model with 2000 dimensions, maximum 13 neighbours per side, 50 decomposition factors and minimum triple occurrence of 5 (see Table 7.31). DTSM is capable to order the similarity groups correctly: the average score for "low" similarity items is smaller than for "medium" and "medium" is smaller than "high" for all three measures (*add*, *mult* and *mwu*).

	add			mult			mwu		
	Mean	SD	SE	Mean	SD	SE	Mean	SD	SE
high	5.12	1.24	0.03	4.11	2.16	0.04	4.57	1.68	0.04
medium	3.83	1.45	0.03	1.88	1.90	0.04	2.71	1.59	0.04
low	3.35	1.47	0.03	1.34	1.74	0.04	2.29	1.56	0.04

Table 7.31: Descriptive statistics for DTSM on Mitchell and Lapata (2010) dataset

We further evaluate the main parameters of the model, similarly to the other experiments, such as:

- context word choice: using words occurring in the corpus 5-50 times as representative for middle frequency words versus using the most frequent words as contexts (Table 7.32);
- context window size: (Table 7.33);
- number of factors (Table 7.34)
- using all three axes of the tensor instead of one (Tables 7.33 and 7.34).

We follow the strategy suggested in the original paper [Mitchell and Lapata, 2010] in that we correlate every single human score for each of the phrases with the system's output; i.e., if participant 1 gave a similarity score of 4 to the sentence pairs *"Figure show increase"* and *"Figure picture increase"*, participant 2 gave a core of 1 and the system assigned 3, then both score pairs ($\langle 4, 3\rangle$ and $\langle 1, 3\rangle$) are evaluated separately for correlation.

Additionally to the *Spearman's correlation* between humans and the system, we report correlation between score1 (addition) and score2 (multiplication) of the components and score3 (*mwu*) of the phrase itself as it is used in the corpus. This way of evaluation was suggested implicitly in Experiment I in Katz and Giesbrecht [2006] and it was explicitly proposed as a means of evaluation of such models by Guevara [2010]. Figure 7.6 visualizes the idea behind this type of evaluation in two dimensions (for simplicity).

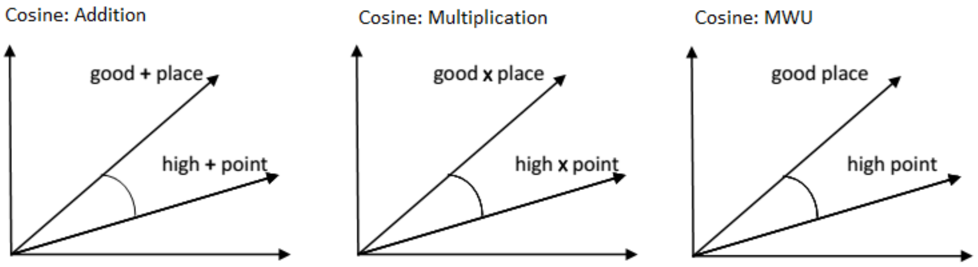

Figure 7.6: Visualization of cosine for *addition*, *multiplication* and *MWU* in two dimensions

This way of evaluating correlation may be a more objective way to measure true performance of the model and compositionality operators as it is independent of the corpus choice and further subjective factors that may have been relevant in constructing the dataset but that are not available for computational systems at this point. An example of such a factor could be world knowledge that is used by human annotators when estimating similarity between the phrases. The automatic system in its current form has only the training corpus (here: **BNC**) as background knowledge.

The conclusions from Tables 7.33 and 7.34 can be summarized as follows:

	all		
	add	mult	mwu
5-50 occurrences	0.23	0.28	0.22
ρ with mwu	0.42	0.48	-
10-10000	0.19	0.27	0.25
ρ with mwu	0.42	0.50	-

Table 7.32: Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $nn = 2$, $f = 50$ and varying context dimension choice

	all			adj-noun			noun-noun			verb-object		
	add	mult	mwu	add	mult	mwu	add	mult	mwu	add	mult	mwu
$nn = 2$	0.19	0.27	0.25	0.35	0.39	0.35	0.21	0.25	0.25	0.12	0.19	0.08
ρ with mwu	0.42	0.50	-	0.40	0.47	-	0.50	0.56	-	0.35	0.37	-
xyz	0.25	0.22	-	0.31	0.30	-	0.27	0.26	-	**0.40**	0.13	-
ρ with mwu	0.37	0.42	-	0.31	0.42	-	0.40	0.37	-	0.38	0.48	-
$nn = 13$	0.33	0.37	0.35	0.39	0.44	0.34	0.37	0.41	0.43	0.29	0.32	0.35
ρ with mwu	0.81	0.88	-	0.84	0.87	-	0.75	0.85	-	0.83	0.90	-
xyz	0.32	0.35	-	0.38	0.37	-	0.37	0.44	-	0.31	0.30	-

Table 7.33: Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $f = 50$ and varying context window size

- for **adjective-noun** pairs: the deployment of further tensor slices other than Y does not bring any improvement with a current model; multiplication is generally better than addition;
- for **noun-noun** phrases: using X and Y slices instead of only Y leads to better precision; multiplication is in overall better than addition for smaller number of factors (50);

	all			adj-noun			noun-noun			verb-object		
	add	mult	mwu	add	mult	mwu	add	mult	mwu	add	mult	mwu
no decomp.	0.30	31	0.32	0.32	0.25	0.35	0.41	**0.49**	0.40	0.22	0.30	0.34
ρ with mwu	0.62	0.68	-	0.63	0.645	-	0.56	0.65	-	0.65	0.72	-
50 factors	0.33	**0.37**	0.35	0.39	**0.44**	0.34	0.37	0.41	0.43	0.29	**0.32**	0.35
ρ with mwu	0.81	0.88	-	0.84	**0.87**	-	0.75	0.85	-	0.83	0.90	-
xyz	0.32	0.35	-	0.38	0.37	-	0.37	0.44	-	0.31	0.30	-
100 factors	0.32	0.35	0.32	0.37	**0.44**	0.35	0.39	0.40	0.40	0.31	0.30	0.34
ρ with mwu	0.80	**0.90**	-	0.80	**0.87**	-	0.74	**0.90**	-	0.80	**0.91**	-
xyz	0.32	0.34	-	0.37	0.38	-	0.39	0.43	-	0.31	0.29	-

Table 7.34: Mitchell and Lapata [2010] dataset: Spearmans ρ for DTSM for $dim = 2000$, $min = 5$, $nn = 13$ and varying factors size

- for **verb-object** constructions: usage of Y and Z slices with a smaller context window ($nn = 2$) improves the results significantly, in all other cases - only marginally; multiplication here is better than addition only when using the middle Y slice for both words and addition is better when using Y and Z slices - here we achieve the precision up to **0.40** which is the best out of all reported in Mitchell and Lapata [2010].
- generally, the bigger context window brings about significantly better results;
- Spearman's correlation with true usages of phrases in the corpus (mwu) is significantly higher for multiplication than for addition;
- there is a very positive correlation between the system's predicted scores and the observed scores of multiword units as they are used in the corpus for all kinds of phrases with a bigger context window ($nn = 13$); the latter confirms the quality of the system's output, independent of the correlation level with human scores.

Table 7.35 presents the comparison of so far DTSM$_{best}$ to the other models from the corresponding paper. DTSM$_{best}$ model consists of 2000 dimensions, maximum 13 neighbours per side and minimum triple occurrences of 5.

The row *"human"* in Table 7.35 presents inter-annotator agreement that was measured by correlating each subject's ratings with the others and then averaging over subjects. Relatively low inter-annotator agreement of 0.49-0.55 reveals the difficulty of the task even for human annotators, which may serve as an indicator that such a dataset may not be the best way to assess the quality of automatic systems. Inter-annotator agreement is usually interpreted as an upper bound for performance of automatic systems.

In general, DTSM variation that is using matrix multiplication and middle matrices performs better than other parameter modifications. It achieves second best result for $adjective - noun$ combinations (together with two reported methods in the paper); its non-decomposed multiplication variant shares the first place for $noun - noun$ phrases with the multiplicative model of Mitchell and Lapata [2010].

A deeper insight into the dataset sheds light on the performance of our model. It catches one's eye, that the notion of similarity is not clearly defined here. Mitchell

model	adjective-noun	noun-noun	verb-object
DTSM$_{best}$	*0.44*	**0.49**	0.32
rho mult-mwu	0.87	0.90	0.91
human	0.52	0.49	0.55
multiplicative model	0.46	**0.49**	0.37
dilation	*0.44*	0.41	0.38
weighted additive	*0.44*	0.41	0.34
target unit	0.43	0.34	0.29
head only	0.43	0.17	0.24
tensor product	0.41	0.36	0.33
additive	0.36	0.39	0.30

Table 7.35: Spearmans ρ for DTSM (dim = 2000, nn = 13, min = 5), models from Mitchell and Lapata [2010] and human similarity ratings

and Lapata [2010] regard in the corresponding paper "semantically equivalent phrases" as the ones that "can be generally substituted for one another in the same context without great information loss" [Mitchell and Lapata, 2010, p. 1407]. Thus, the concept of similarity is vague defined.

For example, a phrase pair *early stage* and *long period* has got a value of similarity 5 (out of 7) from four annotators and equally a value of 1 from another four and further values in-between:

```
Line 56:   participant2 adjectivenouns 0 early stage long period 3
Line 92:   participant3 adjectivenouns 0 early stage long period 5
Line 1315: participant37 adjectivenouns 0 early stage long period 5
Line 1351: participant38 adjectivenouns 0 early stage long period 5
Line 1423: participant40 adjectivenouns 0 early stage long period 1
Line 1495: participant42 adjectivenouns 0 early stage long period 1
Line 1639: participant46 adjectivenouns 0 early stage long period 2
Line 1746: participant49 adjectivenouns 0 early stage long period 2
Line 2431: participant68 adjectivenouns 0 early stage long period 3
Line 2466: participant69 adjectivenouns 0 early stage long period 2
Line 2576: participant72 adjectivenouns 0 early stage long period 3
Line 3403: participant95 adjectivenouns 0 early stage long period 1
Line 4231: participant118 adjectivenouns 0 early stage long period 3
Line 4412: participant123 adjectivenouns 0 early stage long period 5
Line 4519: participant126 adjectivenouns 0 early stage long period 3
Line 4628: participant129 adjectivenouns 0 early stage long period 2
Line 5456: participant152 adjectivenouns 0 early stage long period 1
Line 5527: participant154 adjectivenouns 0 early stage long period 2
```

One more example is *good place* and *high point* that got assigned high similarity values between 4 and 6 by most annotators, but also 1 and 2 by some of them:

```
Line 52: participant2 adjectivenouns 0 good place high point 5
Line 88: participant3 adjectivenouns 0 good place high point 6
Line 1311: participant37 adjectivenouns 0 good place high point 5
Line 1333: participant38 adjectivenouns 0 good place high point 4
Line 1405: participant40 adjectivenouns 0 good place high point 2
Line 1477: participant42 adjectivenouns 0 good place high point 5
Line 1621: participant46 adjectivenouns 0 good place high point 5
Line 1742: participant49 adjectivenouns 0 good place high point 3
Line 2413: participant68 adjectivenouns 0 good place high point 6
Line 2449: participant69 adjectivenouns 0 good place high point 4
Line 2578: participant72 adjectivenouns 0 good place high point 5
Line 3399: participant95 adjectivenouns 0 good place high point 1
Line 4213: participant118 adjectivenouns 0 good place high point 5
Line 4393: participant123 adjectivenouns 0 good place high point 3
Line 4521: participant126 adjectivenouns 0 good place high point 4
Line 4609: participant129 adjectivenouns 0 good place high point 4
Line 5437: participant152 adjectivenouns 0 good place high point 4
Line 5509: participant154 adjectivenouns 0 good place high point 6
```

In both cases, as well as in most other ones in this dataset, "similarity" is rather arbitrary defined: in the case of *"early stage - long period"* it is ontological (belonging to the super-concept of *time (period)*; for *"good place - high point"* it is highly contextual; in *"new situation - different kind"* it may be analogical and so on.

These observations make us believe that measuring the Spearman's correlation between our compositional models (addition and multiplication) and the phrase usage in the corpus as it is (*mwu* in Tables 7.34 and 7.35) is more reliable and

participant20	provide family	home supply	4	HIGH
participant20	provide family	home leave	1	LOW
participant24	provide family	home supply	5	HIGH
participant24	provide family	home leave	1	LOW
participant25	provide family	home supply	5	HIGH
participant25	provide family	home leave	2	LOW

Table 7.36: An example of Grefenstette and Sadrzadeh [2011b] dataset

indicative of the model's quality than the manual scores. The latter values are indeed double as good[1] showing very good correlations.

7.5 Predicting similarity judgments on transitive sentences

Grefenstette and Sadrzadeh [2011b] constructed a dataset of transitive sentences, i.e., sentences consisting of `subjects`, `verbs` and `direct objects` under similar conditions as Mitchell and Lapata [2008, 2010]. The dataset consists of 200 sentence pairs; therefore of a total of 400 sentences.

25 subjects rated each sentence pair for similarity. Table 7.36 shows an excerpt of the dataset. *"Family provide home"* and *"family supply home"* would be an example of a high-similarity pair of sentences; whereas *"family provide home"* and *"family leave home"* is a low-similarity pair.

The task is to let a computational system assign a similarity score to two sentence pairs, similarly to the previous experiments.

7.5.1 Procedure

The **"compositional way"** of treating this problem is similar to the previous exercise for Phrasal Semantics (Section 7.4).

The necessary processing steps include the following:

1. we build the DTSM model from the **BNC** corpus;

[1] Unfortunately, we cannot compare those values to the other systems here.

2. extract word matrices for component words, using two ways of tensor interpretation - only the middle matrix or all three axes;

   For example, for two sentences - *"family provide home"* and *"family supply home"* - that the system should assign a similarity score to, we extract in the first experiment the middle Y matrices of *"family, provide, supply, home"*.

   In the second experiment, X slice is extracted for *subjects* (e.g., *"family"*), Y for *verbs* (e.g., *"provide"* or *"supply"*) and Z for *objects* (e.g., *"home"*). This setting is defined as "XYZ" in Tables 7.37, 7.38 and 7.39.

3. compute addition and multiplication with component word matrices obtained in the previous step;

```
addComp_s1 = matrix(subject1) + matrix(verb1) + matrix(object1);
addComp_s2 = matrix(subject1) + matrix(verb2) + matrix(object1);

multComp_s1 = matrix(subject1) x matrix(verb1) x matrix(object1);
multComp_s2 = matrix(subject1) x matrix(verb2) x matrix(object1);
```

4. compare those by means of cosine;

```
score1(add) = cosine(addComposition_s1, addComposition_s2)
score2(mult)= cosine(multComposition_s1, multComposition_s2)
```

5. calculate Spearman's ρ correlation between the resulting two scores and the gold standard.

7.5.2 Results

Similarly to all previously described experiments, we test a number of parameters for this dataset, such as the number of context dimensions, the minimum number of triple occurrences, the number of decomposition factors, the choice of decomposition method as well as matrix addition and multiplication on non-decomposed matrices.

GS 2011	add	mult	add (direct)	mult (direct)
2000 dimensions	(-0.03)	0.12	(0.02)	0.22
XYZ	(-0.03)	(0.03)	(0.01)	0.12
5000 dimensions	(-0.00)	0.09	(0.02)	**0.24**
XYZ	(-0.01)	(0.02)	(0.0)	(-0.01)

Table 7.37: Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $min = 5$, $nn = 13$, $f = 50$ and varying context dimensions size; insignificant values are in parenthesis (with $p > 0.05$)

GS 2011	add	mult	add (direct)	mult (direct)
50 factors	(-0.00)	0.09	(0.02)	**0.24**
XYZ	(-0.01)	(0.02)	(0.0)	(-0.01)
100 factors	(-0.01)	0.07	(0.02)	**0.24**
XYZ	(-0.02)	(0.03)	(0.00)	(-0.01)

Table 7.38: Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $min = 5$, $nn = 13$ and varying number of decomposition factors; insignificant values are in parenthesis (with $p > 0.05$)

Similarly to the previous experiment, the Spearman's correlation coefficient ρ between human and system's scores is computed following the same strategy; that is, we correlate every single human score for each of the phrases with the system's output; i.e., if participant 1 gave a similarity score of 4 to the sentence pairs *"Figure show increase"* and *"Figure picture increase"*, participant 2 gave a score of 1, and the system assigned 3; then both score pairs ($\langle 4, 3\rangle$ and $\langle 1, 3\rangle$) are evaluated separately for correlation.

The results of this evaluation are in Tables 7.37, 7.38, 7.39 and 7.40.

To sum it up, the best performing setup ($\rho = 0.24$) of DTSM model in this exercise is the one **without decomposition and using matrix multiplication**

GS 2011	add	mult	add (direct)	mult (direct)
$min = 5$	(-0.01)	0.07	(0.02)	**0.24**
XYZ	(-0.02)	(0.03)	(0.00)	(-0.01)
$min10$	(-0.03)	(0.03)	(0.00)	0.16
XYZ	(0.02)	0.08	(0.03)	(0.01)

Table 7.39: Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $nn = 13$, $f = 100$ and varying minimum triple occurrences; insignificant values are in parenthesis (with $p > 0.05$)

GS 2011	add	mult
nmu	(0.00)	0.09
cp_als	(0.02)	(-0.02)
tucker_me	(0.02)	(-0.02)
no decomposition	(0.02)	**0.24**

Table 7.40: Grefenstette and Sadrzadeh [2011b] dataset: Spearman's ρ for DTSM with $dim = 5000$, $min = 5$, $nn = 13$, $f = 50$ and varying tensor decomposition methods; insignificant values are in parenthesis (with $p > 0.05$)

DTSM$_{best}$	**0.24**	Multiply.nmf	0.23
Humans	0.62	Add.svd	0.12
Regression.svd	**0.32**	Verb.svd	0.08
Regression.nmf	0.29	Add.nmf	0.07
Kronecker.nmf	0.25	Verb.nmf	0.04

Table 7.41: DTSM Spearman's correlation compared to state-of-the-art methods [Grefenstette et al., 2013]

as compositionality operator. Using three axes instead of one does not bring any improvement (Tables 7.37, 7.38, 7.39). From all the decomposition methods, the only one that results in positive correlation is the non-negative tensor factorization; still the best result is without using decomposition at all (Table 7.40).

The setup of DTSM that uses **matrix multiplication**, with or without decomposition, is the only one that elicits positive and significant rank correlation.

Table 7.41 shows the performance of the models reported by Grefenstette et al. [2013] versus our best result. **Humans** is inter-annotator correlation. Compared to other models, we are currently 4th placed; but again it is worth mentioning that we just use the model as it is, without any parameter optimization.

8

Summary of Contributions and Outlook

In this thesis, we presented a Distributional Tensor Space Model of natural language semantics. The proposed model offers a solution to two important challenges of the existing state-of-the-art distributional semantics approaches, namely, word order integration and a linguistically plausible compositionality operation.

Existing models that address both issues simultaneously are few. They require advanced preprocessing or sophisticated training of the model parameters. Furthermore, predominating current semantic space models are based on 2nd order tensors, i.e., matrices. Therefore, the amount of information that can be encoded by two describing dimensions is rather limited.

The major question that we pose in this thesis is whether the currently predominating matrix-based semantic space model with vector-based word meaning representation is appropriate as a model of natural language semantics. At least, three issues make us believe that there is a definite need for novel paradigms.

The first challenge we address is the question of word meaning representation. Currently predominating vector-based word semantics seems to be not enough. More and more researchers come to the conclusion that a single vector is too weak to represent word meaning.

The second problem of distributional semantics methods is the integration of word order information into such models. Word order is traditionally taken into consideration in matrix-based approaches by using *HAL*-way of modelling (cf. Chapter 3.2.1), that is by putting only left contexts on one axis and only right contexts on the other axis. Another popular way of treating this topic is by means of using dependency parsed corpora as a basis for the model construction, i.e., they require advanced linguistics preprocessing.

The third important issue is the problem of compositionality of natural language and the lack of appropriate plausible compositionality methods for semantic space models.

As the review of the related literature shows, a number of models has been suggested to solve either the problem of word order integration or the task of compositionality. The models of compositionality including advanced linguistic preprocessing, such as dependency parsing, automatically solve the problem of word order, but require good performance of computational methods for these preprocessing steps. The state-of-the-art accuracy for automatic dependency parsing is below 90% for English and even worse, if available at all, for further languages.

Furthermore, these methods need a lot of training instances per phrase to get reliable results, and usually separate training is required for every kind of expression or combination of those, which is natural language full of.

In this thesis, we suggest a general and intuitive model that offers an all-in-one solution to all three above mentioned topics of concern. **Distributional Tensor Space Model** that is based on 3d order tensors naturally allows to represent words by matrices. We argue that a **matrix-based word representation** allows us to integrate contextual information as well as to model compositionality in a more general manner.

We **solve the problem of word order integration by making use of three dimensions** instead of two that makes it possible to include **triples of information** into the model, which in its turn has a positive co-effect of encoding *three-way dependencies* directly.

The **word representation by means of the matrix** makes its possible, to use matrix multiplication as compositionality operator. We show that the approach of realizing **compositionality via matrix multiplication** is mathematically, linguistically, psychologically and neurologically plausible. Moreover, it was entirely original at the time of publication [Rudolph and Giesbrecht, 2010]; and it is gaining in importance currently.

This *combination of exploiting the advantages of the tensor structure together with word representation as a matrix and usage of matrix multiplication as a compositionality operator constitutes the major novelty of this work.*

Our goal currently is not to get an optimal performance on any certain task. We are looking for a linguistically, mathematically and cognitively adequate model of semantics that is equally suited for most of the semantic processing tasks.

8.1 Conclusions from Evaluation

In order to evaluate the model, we proceeded in the following way. First, the model was assessed on two word similarity benchmarks of distributional semantics (Chapter 6).

The performance of DTSM on these similarity datasets achieves average results. In both cases, the word similarity is rather arbitrary defined and can be better described using the name of one of the datasets as *free associations*. This kind of similarity depends on many external factors, such as social, cultural and so on, so that using a very structured 3d model based purely on the available text collection does not bring any improvement per se.

However, we have detected that DTSM performed better for strongly associated words than the competing semantic space model that was based on latent semantic analysis (LSA) (Chapter 6.1). The LSA model was significantly predominant for random associations, thereby achieving better overall accuracies. The latter makes us believe that matrix-based approaches that do not preserve any text structure are indeed better suited for *random* or *free* similarity tasks.

For Rubenstein dataset (Chapter 6.2) we achieve better results than another state-of-the-art model with advanced preprocessing that is trained on the same training corpus as ours (BNC). Other models have used much bigger corpora, more sophisticated preprocessing and partly other parameter choices, e.g. a different similarity measure than cosine, so that the results are not directly comparable.

We further evaluated DTSM on the task of selectional preferences, one of the advanced natural language processing tasks where word order information matters (Chapter 6.3). Here, DTSM achieves second best result, compared to the ones reported in the corresponding paper, whereas the best model is based on complex preprocessing that is configured specifically for this one task; while our model offers an intuitive simple treatment without any further preprocessing.

Concerning the problem of compositionality, the Distributional Tensor Space Model was evaluated on a number of the standard benchmarks in distributional semantics as well as on self-initiated datasets which may become benchmarks for graded compositionality of phrases and for identification of idiomatic meaning in context.

The conclusions we may draw from our compositionality evaluation campaigns is that DTSM seems to offer a new perspective for sentential composition (Sections 7.2 and 7.5 of Chapter 7). We prove it here mostly by means of very good results for $subject - verb$ composition with a given experimental setup and for given $subject - verb$ combinations. However, we are far from postulating that the suggested model is generally good for all imaginable phrases of any grammatical construction. Semantic compositionality is a too complex concept to be treated by just one universal method in any kind of semantic space. Hence, our findings will be one of the steps towards a better understanding of this task.

For example, as discussed in Chapter 3.3, there are at least two modes of meaning composition: additive and interactive. Until now, these theoretical distinctions have been ignored in the computational models of compositionality. We leave this issue for future research.

To summarize, DTSM achieves good results for important tasks of linguistic semantics by using a relatively small text corpus, without any sophisticated preprocessing and ambitious parameter optimization; unlike the currently predominating models, most of which are optimized or constructed specifically for certain tasks. Moreover, the model is completely language independent.

It is important to keep in mind, that this is a corpus-based approach, and it can be only as good as the corpus is.

The advantage and the elegance of the DTSM model is straightforward; we can use one and the same formalism and operate on 3D-tensors in different ways without any further model transformations: extracting tensor slices, fibres or just values at the intersection of three dimensions.

8.2 Outlook

DiSCo experiments have shown that our model attains very good results for $subject - verb$ constructions; obviously other grammatical constructs follow different rules. Notoriously, there is a number of efforts aimed at learning matrices for $adjective - noun$ units [e.g. Baroni and Zamparelli, 2010; Guevara, 2010]. Thus, our model seems to be complementary to existing functional approaches to composition. This line of further research, together with the extension of the model with the insights from theoretical linguistics, seems to be very promising.

From the computational perspective, we consider "random tensor indexing" - in the style of random indexing methods for matrix space models - as an alternative to currently rather expensive tensor processing and manipulation.

Concerning the Distributional Tensor Space Model itself, further thorough investigations are necessary, especially:

– on the influence of the context dimensions;

– the exact effects of tensor factorization and the number of decomposition factors;

– possibly other matrix similarity measures, apart from cosine.

We were rather limited in our experiments in the possibilities to experiment on the first two points due to restricted computational resources. However, they can be decisive for model's performance.

8.3 Recap of Contributions

Hence, the major contributions of this work can be briefly summarized as follows:

1. **Distributional Tensor Space Model** was proposed as a model of distributional meaning representation;
2. **Matrix consisting of left and right word co-occurrences** was suggested as a means of word meaning representation;
3. **Compositional Matrix Space Model** was defined as a novel type of generic compositional models for syntactic and semantic aspects of natural language, based on matrix multiplication.

References

Alan D. Baddeley. Working memory and language: An overview. *Journal of Communication Disorders*, 36:198–208, 2003. 50

Brett W. Bader and Tamara G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, 32 (4):635–653, 2006. 71

Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.5. Available online, January 2012. URL `http://www.sandia.gov/~tgkolda/TensorToolbox/`. 64

Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 020139829X. 86

Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. An empirical model of multiword expression decomposability. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 89–96, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. 82, 84

Colin Bannard, Timothy Baldwin, and Alex Lascarides. A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 65–72, Sapporo, Japan, 2003. Association for Computational Linguistics. 82, 91

Marco Baroni and Alessandro Lenci. Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010. 30, 31, 32, 33, 43, 73, 74, 75

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, Massachusetts, October 2010. Association for Computational Linguistics. 41, 42, 135

Marco Baroni, Alessandro Lenci, and Luca Onnis. ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 25

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. Frege in space: A program for compositional distributional semantics, 2013. URL `http://clic.cimec.unitn.it/composes/materials/frege-in-space.pdf`. 35, 41, 42

Michael W. Berry, Zlatko Drmac, and Elizabeth R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, June 1999. ISSN 0036-1445. 4

Klinton Bicknell, Jeffrey L. Elman, Mary Hare, Ken McRae, and Marta Kutas. Effects of event knowledge in processing verbal arguments. *Journal of Memory and Language*, 63(4):489–505, November 2010. ISSN 1096-0821. 61, 70, 76, 77

Chris Biemann and Eugenie Giesbrecht. Distributional semantics and compositionality 2011: Shared task description and results. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 21–28, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `ttp://www.aclweb.org/anthology/W11-1304`. ix, xi, 62, 93, 94, 96

Chris Biemann and Valerie Nygaard. Crowdsourcing WordNet. In *Proceedings of the 5th International Conference of the Global WordNet Association (GWC-2010)*, Mumbai, India, 2010. 93

John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907, 2012. 69

Lorna Byrne, Caroline Fenlon, and John Dunnion. IIRG: A naive approach to evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 103–107, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 117

Deng Cai, Xiaofei He, and Jiawei Han. Tensor space model for document analysis. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 625–626, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. 32

J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35(3):283–319, 1970. 14

Arthur Cayley. On the theory of groups as depending on the symbolic equation $\theta^n = 1$. *Philos. Magazine*, 7:40–47, 1854. 54

Peter A. Chew, Brett W. Bader, Tamara G. Kolda, and Ahmed Abdelali. Cross-language information retrieval using PARAFAC2. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 143–152. ACM, 2007. ISBN 978-1-59593-609-7. 32

Stephen Clark and Stephen Pulman. Combining symbolic and distributional models of meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, pages 52–55, 2007. 40

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. Compositional distributional model of meaning. In *Proceedings of the Second Symposium on Quantum Interaction (QI-2008)*, pages 133–140, 2008. 41

Daoud Clarke. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38(1):41–71, 2012. 41

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis*, 36:345–384, 2010. 41, 42

Alan Cruse. *Meaning in Language: An Introduction to Semantics and Pragmatics.* Oxford Textbooks in Linguistics Series. Oxford University Press, 2000. ISBN 9780198700104. 38

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990. 2, 3, 38

Georgiana Dinu, Stefan Thater, and Sören Laue. A comparison of models of word meaning in context. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 611–615, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. 35

David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics.* Reidel, Dordrecht, 1981. 6

Daniel M. Dunlavy, Tamara G. Kolda, , and W. Philip Kegelmeyer. Multilinear algebra for analyzing data with multiple linkages. In Jeremy Kepner and John Gilbert, editors, *Graph Algorithms in the Language of Linear Algebra*, Fundamentals of Algorithms, pages 85–114. SIAM, Philadelphia, 2011. 31

Marc Dymetman. Group theory and computational linguistics. *Journal of Logic, Language and Information*, 7(4):461–497, 1998. 54

Katrin Erk and Sebastian Padó. 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. ACL, 2008. 35, 36

Katrin Erk and Sebastian Padó. Exemplar-based models for word meaning in context. In *Proceedings of the ACL Conference Short Papers*, pages 92–97. Association for Computational Linguistics, 2010. 36

Katrin Erk, Sebastian Padó, and Ulrike Padó. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4): 723–763, December 2010. 76

Stefan Evert. The statistics of word cooccurrences: word pairs and collocations. *Doctoral Dissertation, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart*, 2004. 84

Stefan Evert and Brigitte Krenn. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 188–195, 2001. 20, 61, 84, 88

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. Introducing and evaluating ukWaC, a very large Web-derived corpus of English. In *Proceedings of the WAC4 Workshop (WAC-4)*, 2008. 63, 92

John Rupert Firth. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis*, pages 1–32, 1957. 2

Stefan L. Frank, Mathieu Koppen, Leo G. M. Noordman, and Wietske Vonk. World knowledge in computational models of discourse comprehension. *Discourse Processes*, 45(6):429–463, 2008. 83, 119

Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. TripleRank: Ranking Semantic Web data by tensor decomposition. In *Proceedings of the 8th International Semantic Web Conference*, pages 213–228. Springer-Verlag, 2009. 32

Gottlob Frege. *The Foundations of Arithmetic: A Logico-mathematical Enquiry Into the Concept of Number.* B. Blackwell, 1884. 2, 35

Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based Explicit Semantic Analysis. In *Proceedings of the 20th*

International Joint Conference on Artifical Intelligence, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. 3

Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought.* MIT Press, Cambridge/London, 2004. ISBN 0-262-57219-2. 2

Eugenie Giesbrecht. In Search of Semantic Compositionality in Vector Spaces. In *Proceeding of the 17th International Conference on Conceptual Structures (ICCS)*, pages 173–184, 2009. 8, 38, 83

Eugenie Giesbrecht. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT Student Research Workshop*, pages 23–28. Association for Computational Linguistics, 2010. 8, 31, 43

Jonathan S. Golan. *The theory of semirings with applications in mathematics and theoretical computer science.* Addison-Wesley Longman Ltd., 1992. ISBN 0-582-07855-5. 59

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimenting with transitive verbs in a DisCoCat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. 42

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Stroudsburg, PA, USA, 2011b. Association for Computational Linguistics. xiv, 42, 62, 83, 127, 129, 130

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 125–134, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. 42

Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. Multi-step regression learning for compositional distributional semantics. *ARXIV Preprint:1301.6939*, 2013. xiv, 130

Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Springer, 1994. ISBN 0792394682. 28, 29, 30

Emiliano R. Guevara. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the Workshop on GEometrical Models of Natural Language Semantics*, GEMS, pages 33–37, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. 41, 42, 122, 135

Emiliano R. Guevara. Computing semantic compositionality in distributional semantics. In *Proceeding of the International Conference on Computational Semantics*, pages 135–144, 2011. 41, 42

Mary Hare, Michael Jones, Caroline Thomson, Sarah Kelly, and Ken Mcrae. Activating event knowledge. *Cognition*, 111(2):151–167, May 2009. 76

Zellig Sabbettai Harris. Distributional Structure. In *Word*, volume 10, pages 146–162, 1954. 2

Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16, 1970. 14, 16

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545. Association for Computational Linguistics, 1992. 31

John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979. ISBN 0-201-02988-X. 54

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. UNAL: Discriminating between literal and figurative phrasal usage using distributional statistics and POS tags. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 114–117, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 117

Anders Johannsen, Hector Martinez, Christian Rishøj, and Anders Søgaard. Shared task system description: Frustratingly hard compositionality prediction. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 29–32, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 93

Michael N. Jones and Douglas J.K. Mewhort. Representing word meaning and order information in a composite holographic lexicon. In *Psychological Review*, volume 114, pages 1–37, 2007. 25, 27

Susan Jones and John Sinclair. English lexical collocations. In *Cahiers de Lexicologie*, volume 24, pages 15–61. 1974. 6

Pentti Kanerva. *Sparse distributed memory.* The MIT Press, 1988. 24

Pentti Kanerva, Jan Kristoferson, and Anders Holst. Random Indexing of text samples for Latent Semantic Analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 103–6. Erlbaum, 2000. 24

Graham Katz and Eugenie Giesbrecht. Automatic identification of non-compositional multi-word expressions using Latent Semantic Analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19. Association for Computational Linguistics, 2006. 8, 36, 61, 82, 83, 84, 85, 91, 115, 122

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

Walter Kintsch. Predication. *Cognitive Science*, 25(2):173–202, April 2001. ISSN 03640213. 83, 119

Tamara G. Kolda. Tensor decompositions, the MATLAB tensor toolbox, and applications to data analysis. Available online, April 2007. URL `http://www.ima.umn.edu/industrial/2006-2007/kolda/kolda.pdf`. ix, 14, 15

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. SemEval-2013 Task 5: Evaluating phrasal semantics. In *Second Joint*

*Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics. 62

Joachim Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170, 1958. 54

Joachim Lambek. Type grammar revisited. In *Selected Papers from the Second International Conference on Logical Aspects of Computational Linguistics*, LACL '97, pages 1–27, London, UK, 1999. Springer-Verlag. ISBN 3-540-65751-7. 41

Thomas K. Landauer and Susan T. Dumais. A solution to Plato's problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. In *Psychological Review*, volume 104, pages 211–240, 1997. 2, 3, 38

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Proceeding of the Neural Information Processing Systems (NIPS)*, pages 556–562. MIT Press, 2000. 15

Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. 16

Alessandro Lenci. Composing and updating verb argument expectations: a distributional semantic model. In *Proceedings of the 2Nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 58–66, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. 61, 70, 75, 76, 77, 78, 79

Dekang Lin. Principle-based parsing without overgeneration. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, ACL, pages 112–120, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. 28

Dekang Lin. Principar - an efficient, broad-coverage, principle-based parser. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 482–488, 1994. 28

Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 768–774. Association for Computational Linguistics, 1998. 28, 29, 30, 75

Dekang Lin. Automatic identification of non-compositional phrases. In *Proceedings of the ACL Conference*. Association for Computational Linguistics, 1999. 84

Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien. Text representation: From vector to tensor. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 725–728, Washington, DC, USA, 2005. IEEE Computer Society. 32

Will Lowe. Towards a theory of semantic space. In *Proceedings of the 23rd Conference of the Cognitive Science Society*, pages 576–581, 2001. 7, 19, 20, 22, 23, 28, 33, 45, 47

Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers*, pages 203–220, 1996. 2, 3, 21, 22

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-13360-1. 16, 17, 89

Ken McRae, Michael J. Spivey-Knowlton, and Michael K. Tanenhaus. Modelling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 312(38):283–312, 1998. 75, 76

Ken McRae, Mary Hare, Jeffrey L. Elman, and Todd Ferretti. A basis for generating expectancies for verbs from nouns. *Memory and Cognition*, 33(7):1174–1184, 2005. 76

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244, Columbus, Ohio, 2008. Association for Computational Linguistics. 35, 36, 39, 83, 86, 119, 127

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–429, November 2010. xiii, xiv, 39, 62, 83, 103, 119, 121, 122, 123, 124, 125, 127

Richard Montague. Universal grammar. In Richmond H. Thomason, editor, *Formal Philosophy: Selected Papers of Richard Montague*, number 222–247. Yale University Press, New Haven, London, 1974. 35, 40

Preslav Nakov, Antonia Popova, and Plamen Mateev. Weight functions impact on LSA performance. In *Proceedings of Recent Advances in Natural Language Processing*, pages 187–193, 2001. 33

Preslav Nakov, Elena Valchanova, and Galia Angelova. Towards deeper understanding of the LSA performance. In *Proceeding of Recent Advances in Natural Language Processing*, pages 311–318, 2003. 33

Sebastian Padó and Mirella Lapata. Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, 33(2):161–199, 2007. 28, 29, 30, 45, 73, 74, 75

Tony Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995. 39, 52

Tony A. Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 30–35, 1991. 39

Emil L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946. 57

Liina Pylkknen, Rodolfo Llins, and Gregory L. Murphy. Representation of polysemy: Meg evidence. *JOURNAL OF COGNITIVE NEUROSCIENCE*, 18(1): 1–13, 2006. 37

Gang Qian, Shamik Sural, Yuelong Gu, and Sakti Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1232–1237, New York, NY, USA, 2004. ACM. 17

Reinhard Rapp. Word sense discovery based on sense descriptor dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322, 2003. 69

Siva Reddy, Ioannis Klapaftis, Diana McCarthy, and Suresh Manandhar. Dynamic and static prototype vectors for semantic composition. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 705–713, 2011. 36

Joseph Reisinger and Raymond J Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics, 2010. 36

Philip Resnik. *Selection and Information: A Class-based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1993. UMI Order No. GAX94-13894. 76

Philip Resnik. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 61(1-2):127–159, November 1996. 76

Jennifer M. Rodd, M. Gareth Gaskell, and William D. Marslen-Wilson. Modelling the effects of semantic ambiguity in word recognition. *Cognitive Science*, 28(1): 89–104, 2004. 37

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. An improved model of semantic similarity based on lexical co-occurence. *Communications of the ACM*, 8:627–633, 2006. 22

John B. Rubenstein, Herbert Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8:627–633, 1965. 61, 69, 73

Sebastian Rudolph and Eugenie Giesbrecht. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 907–916, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858681.1858774`. 8, 43, 49, 133

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 1–15. Springer-Verlag, 2002. 81

Magnus Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, 2005. 24, 25, 82, 87

Magnus Sahlgren, Anders Holst, and Pentti Kanerva. Permutations as a means to encode order in word space. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society, CogSci'08*, pages 1300–1305, Washington D.C., USA, 2008. 27, 53

Gerard Salton and Michael J McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York u.a., 1983. 20

Gerard Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. 2, 20

Patrick Schone and Daniel Jurafsky. Is knowledge-free induction of multiword unit dictionary headwords a solved problem? In *Proceedings of Empirical Methods in Natural Language Processing*, 2001. 82, 84

Hinrich Schütze. Dimensions of meaning. In *Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press. 2

Hinrich Schütze. Word space. In *Advances in Neural Information Processing Systems*, volume 5, pages 895–902. Morgan Kaufmann, 1993. 2, 3, 20, 87

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, March 1998. ISSN 0891-2017. 2, 20, 36, 86

Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1-2):159–216, November 1990. 39

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics. 42

Gilbert Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, 1993. 11

Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: Dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 374–383, New York, NY, USA, 2006. ACM. 32

Zoltán Gendler Szabó. Compositionality. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Winter edition, 2012. URL `http://plato.stanford.edu/archives/win2012/entries/compositionality/`. 35

Jakke Tamminen, Alexandra A Cleland, Philip T Quinlan, and M Gareth Gaskell. Processing semantic ambiguity: Different loci for meanings and senses. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, pages 2222–2227, 2006. 37

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957. Association for Computational Linguistics, 2010. 35, 36

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint*

Conference on Natural Language Processing, pages 1134–1143, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. 35

Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3), 1966. 14

Peter Turney. Empirical evaluation of four tensor decomposition algorithms. Technical report, 2007. Technical Report ERB-1152. 32

Peter D Turney. Similarity of semantic relations. *Computational Linguistics*, 32 (3):379–416, 2006. 69

Peter D. Turney. Domain and function: a dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44(1):533–585, May 2012. 43

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010. 4

Tim Van de Cruys. 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, 2009. Association for Computational Linguistics. 32, 33

Tim Van de Cruys. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(04):417–437, October 2010. 32

Tonio Wandmacher, Ekaterina Ovchinnikova, and Theodore Alexandrov. Does Latent Semantic Analysis reflect human associations. In *Proceedings of the Lexical Semantics Workshop at ESSLLI*, Hamburg, Germany, 2008. 72

Dominic Widdows. Semantic vector products: some initial investigations. In *Proceedings of the Second AAAI Symposium on Quantum Interaction*, 2008. 6, 35, 39, 51, 85, 86, 90

Dominic Widdows and Kathleen Ferraro. Semantic vectors: a scalable open source package and online technology management application. *Proceedings of the Sixth International Language Resources and Evaluation (LREC)*, 2008. 64, 87

Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. 1922. 2

Ludwig Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, 1953. 2

George Kingsley Zipf. *The Psychobiology of Language*. Houghton-Mifflin, Boston, 1935. 37