Parameters, Interactions, and Model Selection in Distributional Semantics

Gabriella Lapesa

A thesis submitted in fulfilment of the requirements
for the degree of Ph.D in Cognitive Science
University of Osnabrück, Institute of Cognitive Science
May 2019

Committee
Prof. Dr. Stefan Evert, FAU Erlangen-Nürnberg (supervisor)
Prof. Dr. Kai-Uwe Kühnberger, University of Osnabrück (reviewer)
Prof. Dr. Alessandro Lenci, University of Pisa (reviewer)
Apl. Prof. Sabine Schulte im Walde, University of Stuttgart (reviewer)
Prof. Dr. Jutta L. Müller, University of Osnabrück (chair)

Abstract

Distributional Semantic Models are one of the possible answers produced in (computational) semantics to the question of what the meaning of a word is. The distributional semantic answer to this question is a usage-based one, as distributional semantics models (henceforth, DSMs) are employed to produce semantic representations of words from co-occurrence patterns in texts or documents (Sahlgren, 2006; Turney & Pantel, 2010).

DSMs have proven to be useful in many applications in the domains of Natural Language Processing (Schütze, 1998; D. Lin, 1998), Information Retrieval (Salton et al., 1975), and Cognitive Modeling (Lund & Burgess, 1996; Landauer & Dumais, 1997; Padó & Lapata, 2007; Baroni & Lenci, 2010). Recently, the field of Distributional Semantics has moved towards new challenges, such as predicting brain activation (T. Mitchell et al., 2008; Murphy et al., 2012; Bullinaria & Levy, 2013) and modeling meaning composition (Baroni, Bernardi, & Zamparelli, 2014, and references therein), and towards the use of neural word embeddings (Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013). Despite this progress, a full understanding of the different parameters governing a DSM and their influence on model performance (which, in fact, is also important for getting a better linguistic understanding of neural word embeddings) has not been achieved yet. This is precisely the goal of this dissertation.

Taken together, the experiments presented in this thesis represent (to the best of our knowledge) the largest-scope study in which window and syntax-based DSMs have been tested in all parameter settings. As a further contribution, the thesis proposes a novel methodology for the interpretation of evaluation results: we employ linear regression as a statistical tool to understand the impact of different parameters on model performance. In this way, we achieve a solid understanding of the influence of specific parameters and parameter interactions on DSM performance, which can inform the selection of DSM settings that are robust to overfitting.

This thesis has s strong focus on cognitive data, that is, on DSM parameters that lend themselves to a cognitive interpretation and on evaluation tasks in which DSMs are tested in their capability of mirroring speakers' behavior in psychological tasks (semantic priming and free associations). One of the most important contributions of this thesis is the consistent finding that neighbor rank (i.e., the rank of a word among the distributional neighbors of a target) is a better indicator of semantic similarity/relatedness than the distance in the semantic space, which is commonly used in the literature. The cognitive interpretation of this result is straightforward: neighbor rank, which is evaluated systematically for the first time in this thesis, is able to capture asymmetry in the relation between two words, while distance metrics, commonly employed in distributional semantics, are symmetric.

Acknowledgments

The "making-of" of this dissertation is a warm and crazy mixture of places and people, a journey which would have not been possible without the support I received from multiple fronts.

Thanks to my supervisor, Stefan Evert, for everything he managed to teach me possibly without even realizing it, for his understanding in the dark times, and all the fun we had and still have while working together (in particular with late-night deadlines approaching).

Thanks also go to Sabine Schulte im Walde for taking care of me and for her endless attempts to push my progress, often the hard way, but also with a hug when needed, and to Alessandro Lenci, who taught me what computational linguistics was and basically never got rid of me. I am also grateful to Kai-Uwe Kühnberger for agreeing to review my thesis (together with Stefan, Sabine, and Alessandro) and for his support in the organization of my defense. Thanks go also to Jutta Müller for chairing it.

My defense happened in Osnabrück, in December 2019, at a point in time when I did not live there anymore. Not only did a large part of the committee take a trip to the cold north (extra thanks for that, Alessandro, Sabine, and Stefan), but I have also been joined for the celebration by an amazing group of friends. Thanks so much for being there with me, Agnieszka, Diego, Enrica, Martin, Sascha, and Thomas! I would also like to thank Christian, Diego, Enrica, and Sascha for proof-reading the thesis.

Through the years, I had the luck of working in different institutes, research groups, and cities. This implied establishing a very intimate relationship with *Umzugkartons*, but also gave me a chance of meeting many wonderful people.

It all started in Osnabrück, at the Institute of Cognitive Science (IKW), where I have been a PhD student thanks to a Georg-Christoph-Lichtenberg scholarship granted by the Ministry for Lower Saxony. At the IKW, I have been a member of the Computational Linguistics group, headed by Peter Bosch: thanks, Peter, for your support, advice, and triggering so many lively discussions. Thanks also go to all the group members: Martin Aher, Sascha Alexeyenko, Maria Cieschinger, Stefan Evert, Stefan Hinterwimmer, Mingya Liu, Mikko Määttä, Umesh Patil, and Carla Umbach. The IKW years are full of warm memories: Cristina's constant support and all the fun adventures we shared; Marta's love packed in Netto bags; Hazem's unstable furniture; Ian's underspecified recipes; Mikko's logical arguing; Anna's warming smiles; Martin, our dinners, the silly movies, and your very concrete ways of caring and always keeping an eye on me.

Later on, I was lucky to be awarded a DFG scholarship to spend a year in the Integrated Graduate School of the SFB 732 at the Institute for Natural Language Processing (IMS) in Stuttgart. There, I was a member of the Distributional Approaches

to Semantic Relatedness group, working with Sabine Schulte im Walde. The IMS environment has really been crucial in my development as a researcher, starting from that first stay up to now. I would not have known about that scholarship, as important as it was for my future, if my "old" friend Alessandra hadn't pointed it out to me: thanks, Ale, for this and for always taking care of me, from close and from far.

Next stop was FAU Erlangen-Nürnberg and the Corpus Linguistics group, which was (and still is) family to me. Thanks to: Besim, for the train rides and all the chocolate; Thomas, for all your help with the parsers and the HPC and for your refreshingly fun sarcasm; Paul, for your warm kindness; Philipp, you joined only during my last months, but it feels you have been there from the beginning, too; Peter, for coming to my rescue, many times, always with a smile.

Then was IMS Stuttgart again, in the Theoretical Computational Linguistics group headed by Sebastian Padó. Thanks, Sebastian, for your support and patience with a "certain delay", for the puns, and for being always such a productive and fun thinking partner. Thanks also go to all the TCL members throughout the years. IMS has also brought me new friends: Christian, "just an office neighbor", who ended up being one of my best friends – thanks for being there when I need you, even from far; Enrica, who kept my mood up squirrelling around in the office; Aga, who is never short of hugs; Jana, partner in crime. Diego, I don't know whether I should thank you for all the delicious cooking and the weight we both gained since our paths rejoined in Stuttgart, but one thing is sure: you always have my back, since when I know you, thank you so much for this! Through the PhD rollercoaster, the rest of my Pisa "crowd" has also never stopped being there for me: thanks, Raffaele, Martina, Vins, Maria, Andrea.

At different stages of this dissertation, many other people were also supporting and encouraging me. Among them, I would especially like to thank Ken McRae (who also hosted me in his lab for a research stay), Carla Umbach, Nicholas Asher, Ingo Plag, Dan Jurafsky, Hans Kamp, and Jonas Kuhn.

Thousand thanks go my family, who encouraged and supported me through all my studies, even if these brought me far away from them.

And finally, infinite thanks go to Sascha for his love, his stubborn support, and for growing up together with me through all we shared.

Contents

Abstract iii

Acknowledgments v

1 Introduction 1
- 1.1 Contributions . . . 2
- 1.2 Thesis overview . . . 3
- 1.3 Publications . . . 4

2 Introducing Distributional Semantic Models 7
- 2.1 Motivation . . . 9
  - 2.1.1 Linguistics as Mathematics: Structuralism and early Corpus Linguistics . . . 9
  - 2.1.2 Distributional Hypothesis and first distributional models . . . 11
  - 2.1.3 The linguist's view on DSMs . . . 13
    - 2.1.3.1 DSM representations and word polysemy . . . 14
    - 2.1.3.2 Semantic similarity in DSMs is too underspecified . . . 14
    - 2.1.3.3 DSMs and reference . . . 15
    - 2.1.3.4 DSMs and entailment . . . 16
    - 2.1.3.5 DSMs and non-distributional semantic knowledge . . . 17
    - 2.1.3.6 DSM representations above the word level . . . 17
  - 2.1.4 The grounding problem . . . 19
- 2.2 Formal definitions of DSMs . . . 20
- 2.3 DSM parameters: a taxonomy . . . 25
  - 2.3.1 Corpus selection and pre-processing . . . 25
  - 2.3.2 Extraction: from a corpus to a matrix . . . 28
    - 2.3.2.1 Context selection function . . . 29
    - 2.3.2.2 Path value function . . . 33
    - 2.3.2.3 Target and basis mapping functions . . . 35
    - 2.3.2.4 Quantifying co-occurrences . . . 38
  - 2.3.3 Manipulation of the co-occurrence matrix . . . 41
    - 2.3.3.1 Feature weighting and transformation . . . 42
    - 2.3.3.2 Dimensionality reduction . . . 47
  - 2.3.4 Projecting meaning in space . . . 52
    - 2.3.4.1 Distance measures . . . 52
    - 2.3.4.2 Relatedness in the semantic space . . . 55
- 2.4 DSM representations based on signal vectors . . . 56

2.4.1 Count DSMs based on signal-vectors . . . 58
2.4.2 Predict DSMs based on signal vectors . . . 60
2.4.3 Signal-vectors vs. co-occurrence based DSMs . . . 63

3 Evaluation of DSMs 65
3.1 Classification criteria . . . 66
3.2 Semantic similarity tests . . . 67
3.2.1 Multiple-choice (synonymy) test . . . 67
3.2.2 Prediction of similarity ratings . . . 68
3.2.3 Clustering . . . 68
3.3 Cognitive modeling . . . 69
3.3.1 Prediction of free association norms . . . 69
3.3.2 Priming: modeling of reaction times . . . 70
3.4 Multiple choice on priming datasets . . . 71

4 Experimental setting 73
4.1 Corpus selection and pre-processing . . . 73
4.2 Extraction: from a corpus to a matrix . . . 79
4.2.1 Context selection function . . . 79
4.2.1.1 Surface-based co-occurrences . . . 81
4.2.1.2 Dependency-based co-occurrences . . . 81
4.2.2 Path value, basis mapping, and co-occurrence quantification . . . 85
4.3 Manipulation of the co-occurrence matrix . . . 86
4.3.1 Dimensionality reduction . . . 87
4.4 Projecting meaning in space . . . 88
4.5 Selection of word similarity tasks . . . 89
4.6 Computational tools . . . 90
4.7 Summing up . . . 91

5 Interpreting DSM performance 93
5.1 Standard approaches . . . 94
5.2 Interpreting performance with linear regression . . . 96
5.3 In practice . . . 97
5.4 Summing up . . . 104

6 Evaluation of window-based DSMs: Word similarity tasks 107
6.1 TOEFL . . . 108
6.1.1 Best parameter values . . . 110
6.2 Similarity ratings . . . 119
6.2.1 Best parameter values . . . 121
6.3 Clustering . . . 128
6.3.1 Best parameter values . . . 129
6.4 Index of distributional relatedness . . . 137
6.5 Best settings . . . 139
6.6 Summing up . . . 140

7 Syntax-based DSMs: Are they worth the effort? 143
7.1 Feature ablation . . . 144
7.2 Dependency filtered models . . . 147
7.3 Dependency typed models . . . 152
7.4 Index of distributional relatedness . . . 155
7.5 Best settings . . . 156
7.6 Summing up . . . 158

8 Modeling syntagmatic and paradigmatic relations 159
8.1 Previous work . . . 160
8.2 Multiple choice task on priming datasets . . . 161
8.2.1 Best parameter values . . . 164
8.2.2 Best settings . . . 169
8.2.3 Dependency-based models . . . 172
8.3 Reverse free association task . . . 174
8.3.1 Experimental setup . . . 175
8.3.2 Results . . . 177
8.4 Summing up . . . 179

9 Conclusion 181
9.1 Further work on modeling reaction times in priming . . . 183
9.2 Future steps . . . 184

A Clustering implementation: `pam` vs. CLUTO 187

B Best Models 189

C Distribution of Performance 201

D NaDiR: implementation details 215

1

Introduction

Distributional Semantic Models are one of the possible answers produced in (computational) semantics to the question of what the meaning of a word is. The Distributional Semantic answer to this question is a usage-based one: distributional semantics models (henceforth, DSMs) are employed to produce semantic representations of words from co-occurrence patterns in texts or documents (Sahlgren, 2006; Turney & Pantel, 2010).

Building on the Distributional Hypothesis (Harris, 1954; Miller & Charles, 1991), DSMs quantify the amount of meaning shared by words as the degree of overlap of the sets of contexts in which they occur. A widely used approach operationalizes the set of contexts as co-occurrences with other words within a certain window. **Window-based** DSMs (also known as *bag-of-words*, *term-term*, or *vanilla*) can be represented as a co-occurrence matrix in which rows correspond to target words, columns correspond to context words, and cells store the co-occurrence frequencies of target words and context words. Alternatively, DSMs can be constructed relying on the occurrence patterns of target words in sentences or documents. **Document-based** (or *term-document*) DSMs can be represented as an occurrence matrix in which rows correspond to target words, and columns correspond to documents or sentences. Taken together, term-term and term-document models are an instance of an approach to the collection of co-occurrence which is based on accumulation of co-occurrence counts. An alternative approach adopts neural network architectures which are trained in the task of predicting co-occurrences and in doing so learn dense and low-dimensional distributional representations called **neural embeddings** (Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013).

DSMs have proven to be useful in many applications in the domains of Natural Language Processing (Schütze, 1998; D. Lin, 1998), Information Retrieval (Salton et al., 1975), and Cognitive Modeling (Lund & Burgess, 1996; Landauer & Dumais, 1997; Padó & Lapata, 2007; Baroni & Lenci, 2010). Recently, the field of Distributional Semantics has moved towards new challenges, such as predicting brain activation (T. Mitchell et al., 2008; Murphy et al., 2012; Bullinaria & Levy, 2013) and modeling meaning composition (Baroni, Bernardi, & Zamparelli, 2014, and references therein), and towards the use of neural word embeddings (Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013). Despite this progress, a full understanding of the different parameters governing a DSM and their influence on model performance (which, in fact, is also important for getting a better linguistic understanding of neural word embeddings) has not been achieved yet. This is precisely the goal of this dissertation:

it introduces a novel evaluation methodology and discusses the results of its application to a large-scale evaluation study of DSMs.

Scope When it comes to DSM evaluation, there are always more parameters and tasks to explore. Evaluation studies, however, face practical issues and the choice of parameters (e.g., window size) and parameter values (e.g., one, ten, twenty words) is determined by the interplay of many factors, for example: practical considerations concerning what is computationally feasible; the state of the art in the field and the designer's feeling of what is yet to be explored or deserves a more thorough exploration; the nature of the datasets to be modeled. With this in mind, let us proceed to define the scope of this thesis.

As far as the class of the evaluated DSMs is concerned, we focus on window-based and syntax-based DSMs and do not consider document-based DSMs and neural embeddings. Document-based DSMs fall out of the scope of this work because of their limited parameter space (they provide no room for DSM parametrization as far as the extraction of co-occurrence information is concerned) and the fact that, albeit popular in early Distributional Semantic work (Landauer & Dumais, 1997), the focus of the research community has now shifted to term-term DSMs. Further, window-based and syntax-based DSMs have taken priority over their neural-embedding counterparts for three main reasons. First, neural embeddings lack interpretable dimensions, which is a crucial limitation when dealing with tasks or applications that rely on interpretable distributional features. Second, it has been shown in the literature that some successful word embeddings are indeed mathematically equivalent to a term-term DSM (Levy & Goldberg, 2014b) – so the results from this dissertation also apply to embeddings to some degree. Third, in the field of neural embeddings, new architectures keep on being devised, often as slight variations of previous models, which makes it difficult to identify a clear-cut parameter space.

Readership Given that distributional semantics is situated at the interface of various disciplines, the potential readership of this dissertation goes beyond the NLP community. In particular, it should be of interest for scholars in Cognitive Science (due to its focus on cognitive datasets and cognitive parameters), Corpus Linguistics (as its findings contribute to a better understanding of the association measures commonly employed to detect collocations), as well as Theoretical Linguistics (since a solid understanding of the underlying representations of distributional models is the only way for them to be used for addressing theoretical issues).

1.1 Contributions

The main contributions of the present thesis are at multiple levels, which are spelled out below.

A large scale evaluation study involving all parameter combinations. Taken together, the experiments presented in this thesis represent (to the best of our knowledge) the largest-scope study in which window and syntax-based DSMs have been tested in all parameter settings. While this goal required a significant computational effort, it also allowed to draw robust conclusions – in particular in combination with the proposed evaluation methodology.

A novel methodology for interpreting DSM performance. We employ linear regression as a statistical tool to understand the impact of different parameters on model performance. DSM parameters and their interactions are considered predictors of model performance. In this way, we achieve a solid understanding of the impact of specific parameters and parameter interactions on DSM performance, which can inform the selection of DSM settings that are robust to overfitting.

Cognitive tasks and cognitive parameters. In this thesis, a strong focus is put on cognitive data, that is to say, on DSM parameters that lend themselves to a cognitive interpretation and on evaluation tasks in which DSMs are tested in their capability of mirroring speakers' behavior in psychological tasks (semantic priming experiments and generation of free associations). One of the most important contributions of this thesis is the consistent finding that neighbor rank (i.e., the rank of a word among the distributional neighbors of a target) is a better indicator of semantic similarity/relatedness than the distance in the semantic space, which is commonly used in the literature. This is particularly true for the aforementioned cognitive tasks – which is not surprising, given the capability of neighbor rank to capture asymmetric relations between words.

1.2 Thesis overview

In what follows, we provide an overview of the thesis by presenting a short summary of the content of each chapter.

Chapter 2 (*Introducing distributional semantic models*) sets up the stage of this dissertation. It starts by outlining the history and the motivation of Distributional Semantics as an empirical methodology and proceeds to review the linguistic and cognitive desiderata for Distributional Semantics as a theory of meaning. Then the chapter takes a formal turn and frames its description of DSM parameters into a formal definition of distributional semantic models which extends previous proposals in the literature. The chapter concludes by reviewing alternative ways to build distributional representations from co-occurrence data.

Chapter 3 (*Evaluation of DSMs*) reviews existing tasks proposed for DSM evaluation. The chapter opens by discussing criteria for a taxonomy of DSM evaluation tasks and then zooms in on word-level similarity and cognitive modeling, which are in the focus of the thesis.

Chapter 4 (*Experimental setting*) defines the scope of the experiments presented in the thesis and provides all details concerning evaluated parameters, evaluation tasks, and computational tools employed to carry out the experiments.

Chapter 5 (*Interpreting DSM performance*) describes the linear regression approach to the interpretation of DSM performance which is proposed in this thesis. In doing so, it motivates the features of the proposed methodology with respect to previous approaches in the literature and provides the technical coordinates to interpret the results and the plots presented in the following chapters.

Chapter 6 (*Evaluation of window-based DSMs: Word similarity tasks*) presents the results of the evaluation of window-based DSMs on word similarity tasks (multiple choice

synonymy task, prediction of similarity ratings, noun clustering). The fine-grained discussion provided in this chapter is also meant as a hands-on guideline for the application of the regression methodology described in chapter 5.

Chapter 7 (*Syntax-based DSMs: Are they worth the effort?*) complements the window-based experiments in chapter 6 with evaluation of syntax-based DSMs on the same tasks.

Chapter 8 (*Modeling syntagmatic and paradigmatic relations*) turns to cognitive datasets and presents the results of experiments on a multiple-choice task based on priming datasets and on the prediction of free association norms. Additionally, it explores to what extent the results obtained on similarity tasks carry over to cognitive tasks.

Chapter 9 (*Conclusion*) summarizes the main findings and contributions of the thesis and describes potential future research directions. In addition, it provides an overview of further work in the domain of cognitive modeling which has been conducted within the frame of the dissertation.

1.3 Publications

The output of the research described in this thesis integrates and extends the results which have been presented in a number of earlier publications and conference presentations. The list of these publications and presentations is given below.

- Lapesa, G., & Evert, S. (2017). Large-Scale Evaluation of Dependency-Based DSMs: Are They Worth the Effort? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (pp. 394–400). Valencia, Spain.

- Lapesa, G., & Evert, S. (2014). A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions, and Model Selection. *Transactions of the Association for Computational Linguistics*, 2, 531-545.

- Lapesa, G., & Evert, S., & Schulte im Walde, S. (2014). Contrasting Syntagmatic and Paradigmatic Relations: Insights From Distributional Semantic Models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)* (pp. 160–170). Dublin, Ireland.

- Gabriella Lapesa, Stefan Evert (2014). NaDiR: Naive Distributional Response Generation. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)* (pp. 50-59). Dublin, Ireland.

- Lapesa, G., & Evert, S. (2013). Evaluating Neighbor Rank and Distance Measures as Predictors of Semantic Priming. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)* (pp. 66-74). Sofia, Bulgaria.

- Lapesa, G., Schulte im Walde, S., & Evert, S. (2014). *Judging Paradigmatic Relations: A Collection of Ratings for English.* Poster presented at the Architecture and Mechanisms of Language Processing conference (AMLAP-2014). Edinburgh, UK.

- Lapesa, G., & Evert, S. (2013). *Thematic Roles and Semantic Space. Insights from Distributional Semantic Models.* Paper presented at the Quantitative Investigations in Theoretical Linguistics conference (QITL-5). Leuven, Belgium.

- Lapesa, G., & Evert, S. (2013). *Item-based Prediction of Reaction Times in Priming: an Evaluation of Distributional Semantic Models.* Poster presented at the Architecture and Mechanisms of Language Processing conference (AMLAP-2013). Marseille, France.

2

Introducing Distributional Semantic Models

Distributional Semantic Models (henceforth, DSMs) are computational models employed to produce semantic representations of words from co-occurrence patterns in texts or documents (Sahlgren, 2006; Turney & Pantel, 2010). Building on the Distributional Hypothesis (Harris, 1954; Miller & Charles, 1991), DSMs quantify the amount of meaning shared by words in terms of the degree of overlap between the sets of contexts in which they occur.

A DSM can be considered as a collection of lexical entries which aims at covering a certain proportion of words in a language. A DSM lexical entry for a **target** word (e.g., *student*) is neither a descriptive definition of the type we would find in a dictionary (e.g., *A person who is studying at a university or other place of higher education*[1]), nor a set-theoretic definition of the type we would encounter in formal semantics (e.g., $\lambda x.\textsc{student}(x)$, the set of all x's of which it is true that x is a student).

DSM lexical entries are tuples of numerical values (i.e., **vectors**) of the type $student = (5, 0, 100)$. Every position in a lexical entry is related to a specific feature which characterizes the meaning of the target word, while the corresponding numerical value quantifies the extent to which that feature is representative for the target word. In more detail, the features contained in DSM lexical entries (also referred to as **contexts**) usually correspond to the words which are mentioned together with the target in a collection of texts (e.g., for our *student* example, w_1 = *adolescent*; w_2 = *flour*; w_3 = *university*), but they can also correspond to documents in which the target occurs (e.g., d_1 = an article about education; d_2 = a pizza recipe; d_3 = a university leaflet). At this point, the reader may ask why DSM lexical entries (henceforth, **distributional vectors**) need to contain zeros, or, in other words, why do we need to keep track of the absent contexts (in our example, the second position of the *student* vector, corresponding to w_2 in the words-as-contexts DSM and to d_2 in the documents-as-contexts DSM). The reason is that distributional semantic modeling is usually not concerned with the meaning of one word (e.g., *student*), but aims at quantifying **similarity** between words (e.g., *student* vs. *baker*). To ensure vector comparability, all lexical entries in a DSM need to contain information about all possible features (all the feature words or all the documents in the text collection). Needless to say, the frequency information needs to be stored in the same order for all distributional vectors in the same DSM (e.g., $student = (w_1 = 5, w_2 = 0, w_3 = 100)$, $baker = (w_1 = 2, w_2 = 100, w_3 = 2)$). By definition, a DSM is a collection of vectors: a **matrix**. Distributional vectors can also

[1]Definition from the Oxford Advanced learner's Dictionary.

be seen as the coordinates of points in a multidimensional space which has as many dimensions as there are positions in the vectors (in our examples, we would be dealing with a three-dimensional space). Since each word in the DSM is identified as a point in the same space, the semantic similarity between words can be quantified by calculating the distance between the corresponding points; for a review of the geometric metaphor of meaning, see Widdows (2004).

In the previous paragraph we sketched the two main approaches to the collection of co-occurrence information for a DSM, which we defined, for the sake of introductory simplicity, words-as-contexts and documents-as-contexts approach. In the DSM literature, words-as-contexts DSMs are usually known as **term-term, bag-of-words, or window-based** (because co-occurrence information is extracted by making reference to a specific window-size, e.g., 5 words to the left and to the right of the target), while documents-as-contexts DSMs are referred to as **term-document or document-based**.[2]

Once the DSM matrix has been collected, a number of operations can still be performed on it, in order to improve the semantic representations or to make the matrix more manageable. One example of an issue addressed by matrix manipulation is high-frequency bias: contexts words with high frequency or very long documents tend to occur with many target words, making target vectors in which they occur more similar than desired. To smoothen such bias, vectors can be weighted by employing a number of association measures (Evert, 2008), while a number of mathematical transformations can be applied to scored vectors in order to reduce the skewness of co-occurrences (see section 2.3.3.1 for concrete examples of weighting and transformation operations). Since co-occurrence matrices tend to be very large and sparsely populated, dimensionality reduction techniques are often used to obtain more compact representations. Landauer & Dumais (1997) claim that dimensionality reduction also improves the semantic representation encoded in the DSM matrix, because it captures latent relations between context dimensions and reduces the noise present in the co-occurrence data.[3] Finally, distances between the row vectors of the matrix are computed and – according to the Distributional Hypothesis – interpreted as a correlate of the semantic similarities between the corresponding target words.

Literature shows that different design choices can lead to quite different similarities for the same words (Sahlgren, 2006): the aim of this chapter is to provide a general introduction to DSMs by reviewing the many parameters involved, as well as their implications for the resulting semantic representations.

Section 2.1 discusses the theoretical motivations of Distributional Semantics with respect to linguistic theory, psychology, and cognitive modeling, and it reviews the different sources of criticism to the view of DSMs as linguistic and cognitive models of semantic knowledge. Section 2.2 starts by reviewing some formal definitions of DSMs proposed in the literature, and then updates them to account for the recent develop-

[2] A terminological alternative is proposed by Sahlgren (2006), and is based on the type of relations among words that a specific type of model is able to capture. Based on a number of evaluation experiments, Sahlgren defines term-document models *syntagmatic*, and the term-term ones *paradigmatic*. This classification is not adopted in this thesis, for two reasons: first, the syntagmatic vs. paradigmatic distinction pertains to a different level (interpretative and not descriptive); second, the experiments reported in chapter 8 show that the picture is not as clear cut as depicted in Sahlgren's experiments.

[3] Note that feature weighting, transformations on the counts, and dimensionality reduction are not applied to neural embeddings.

ments in the field of Distributional Semantics. Relying on the theoretical framework sketched in section 2.2, section 2.3 describes in detail the steps necessary to build a DSM along with the parameters associated with every step. Finally, section 2.4 reviews alternative approaches to the extraction of distributional representations, including, but not limited to the neural-network approach for the extraction of neural embeddings (Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013).

2.1 Motivation

In this section we review the motivation of Distributional Semantics as a research field, with a focus on the evolution of theoretical and psychological approaches to distributional semantic representations. The section is structured chronologically. The review of the evolution of theoretical and psychological theories is paired with a brief description of the corpus-based work corresponding to each conceptual step in the development of the Distributional Hypothesis. We conclude by summarizing the main problematic issues about the nature of the semantic representations encoded in DSMs.

In the 1950s, the intuition that the statistical distribution of linguistic elements can be a powerful tool for the investigation of the structure of language gave rise to an innovative methodology developed in the Structuralist framework and primarily focussed on morphology and phonology (Harris, 1954); in the same years, the early Corpus Linguistics tradition (Firth, 1957) proposed to exploit statistical distribution to characterize word meaning: "You shall know a word by the company it keeps."(Firth, 1957). Section 2.1.1 provides a review of the early distributional analyses for the facts of language.

Section 2.1.2 discusses how, in the 1990s, the distributional methodology devised in the Structuralist framework turned into a full-fledged theory of the representation and acquisition of word meaning. Among the factors which determined such evolution are the development of usage-based theories in psychology and psycholinguistics (Miller & Charles, 1991) and the progress of distributional models for information retrieval (Salton et al., 1975; Deerwester et al., 1990; Schütze, 1992) and cognitive modeling (Lund & Burgess, 1996; Landauer & Dumais, 1997).

Section 2.1.3 and 2.1.4 review different points of criticism moved to DSMs. Section 2.1.3 is concerned with the criticism coming from the Theoretical Linguistics side, and it sketches the progress triggered by the need of addressing theoretical issues. Section 2.1.4 focuses on the most problematic aspects connected with the status of DSMs as cognitive models.

2.1.1 Linguistics as Mathematics: Structuralism and early Corpus Linguistics

Zelig Harris' *Distributional Structure* (Harris, 1954) is credited as the first proposal for the use of distribution as a scientific procedure to investigate word meaning. Even if this is fully motivated from a methodological point of view, the main focus of *Distributional Structure* is within the fields of morphology and phonology (word meaning being only marginally touched upon as a further application), and the meaning of morphemes and phonemes as discrete categories is a somewhat different entity from the meaning of words that Distributional Semantics is concerned with.

Before Harris, the need for a mathematical linguistics had been advocated by Martin Joos: "We must adopt a technique of precise treatment, which is by definition a mathematics. **We must make our 'linguistics' a kind of mathematics, within which inconsistency is by definition impossible**." (Joos, 1950, p. 702). In particular, linguistics was identified with discrete mathematics: this implied that all phenomena that are continuous in nature had to be kept out of the scope of linguistic analysis[4]: "All continuity, all possibilities of infinitesimal gradation, are shoved outside of linguistics in one direction or the other. There are in fact two such directions in which we can and resolutely do expel continuity: semantics and phonetics."(Joos, 1950, p. 705). In Joos view, the meaning of a morpheme corresponds to "the set of conditional probabilities of its occurrence in context with all other morphemes – of course without inquiry into the outside, practical, or sociologist's meaning of any of them."(Joos, 1950, p. 708). Interestingly, the possibility of a "structural semantics" was therefore not completely ruled out, but made conditional to further mathematical transformations on the set of conditional probabilities (the largest probabilities replaced by one and the smaller ones by zero) to satisfy the discreteness assumption.

Zelig Harris defines the distributional structure of language as the **description of the occurrence of linguistic elements relative to other linguistic elements** (Harris, 1954)[5], and he devises an **empirical methodology** firmly grounded in the Structuralist tradition (Saussure, 1916). In the Structuralist view, linguistic signs have no value independent of the system of oppositions defined by the language system to which they belong: linguistic signs can only be defined *differentially*, i.e., by describing their differences from other linguistic signs within the same system. In Harris's view, a distributional methodology can make a twofold contribution to the structuralist description of the facts of language. First, the quantification of relative occurrence between items allows the identification of those items that can be considered constitutive elements of the linguistic system under investigation (e.g., a set of phonemic or morphemic solutions for a given language). Second, the description of the relative occurrence of linguistic elements allows the quantification of the similarities (and differences) holding among them, placing the items in a network of relations, as an empirical implementation of the differential approach to linguistic analysis.

A number of arguments are provided, which support the application of distributional methodology for language analysis:

- distribution is a **constitutive** feature of language: when we speak, we do not displace linguistic items arbitrarily with respect to each other;
- distribution allows for **gradedness**, providing a description of language facts that

[4] "The linguistic categories are absolutes which admit of no compromise. [...] The correspondence between the discrete categories of the language and the continuous phenomena of the real world is not and cannot be precise. Our reaction, as linguists, to this situation is very simple: all phenomena, whether popularly regarded as linguistic (such as the tone of anger in an utterance) or not, which we find we cannot describe precisely with a finite number of absolute categories, we classify as non-linguistic elements of the real world and expel them from linguistic science. Let sociologists and others do what they like with such things – [...] – in a word, for us they represent that 'continuity' which we refuse to tolerate in our own science."(Joos, 1950, p. 705)

[5] "The distribution of an element will be understood as the sum of all its environments. An environment of an element A is an existing array of its co-occurrents, i.e. the other elements, each in a particular position, with which A occurs to yeld an utterance. A's co-occurrents in a particular position are called its selection for that position."(Harris, 1954, p. 775)

is both exact (e.g., the relative co-occurrence of class members can be stated exactly) and flexible (e.g., the membership of elements to a specific class can be defined in terms of frequency or probability);

- distribution is a **self-sufficient** criterion for the description of linguistic data: relative position of linguistic elements is an empirical fact, that can always be established;
- distribution allows for **mathematical generalization**: this property enables the researcher to identify less complex systems for the explanation of large amounts of data.

Similarly to what proposed by Joos and in accordance to the Structuralist tradition, meaning is discarded from the pool of potential explanatory factors for the facts of language: "As Leonard Bloomfeld pointed out, it frequently happens that when we do not rest with the explanation that something is due to meaning, we discover that it has a formal regularity or 'explanation'." (Harris, 1954, p. 785). Meaning cannot satisfactorily explain linguistic facts because it is not a unique property of language but rather a general characteristic of human activity: as a consequence, we cannot rely on a one-to-one relation between language and meaning. However, even if meaning cannot play the role of the *explanans* in linguistic analysis, it can still be on the *explanandum* side. That part of meaning which surfaces in language can be described in terms of the distributional regularities with which it correlates: "... if we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, **difference of meaning correlates with difference of distribution**." (Harris, 1954, p. 776).

In the same years, within Computational Linguistics, the sub-discipline of Corpus Linguistics originated from the Firthian notion of "habitual collocations"(Firth, 1957): word meanings can be characterized in terms of a ranked list of other word with which they have a strong tendency to co-occur (i.e., the contexts of its common usage). The concerns of Corpus Linguistics were focussed on language documentation and lexicography, and corpus-based lexicographers went that far in claiming that word senses do not exist as an abstract entity, but they can only be gathered from the observation of concrete examples of word usage (Kilgarriff, 1997). While Corpus Linguistics shared with the Structuralist tradition the empirical methodology based on contextual information and a general agnosticism concerning what meaning is outside of linguistic utterances, the main element of novelty was the fact that words and their meanings were posited as the basic object of interest of the linguistic analysis.

2.1.2 Distributional Hypothesis and first distributional models

While Harris's *Distributional Structure* constitutes the methodological foundation of Distributional Semantics, the *Contextual Hypothesis* elaborated by George Miller and Walter Charles in the '90s provided the psychological argument for a usage-based characterization of meaning on the basis of distributions. The Contextual Hypothesis integrates the notion of semantic similarity (a structuring principle of the semantic memory in the psychological tradition) with the associative learning mechanisms described in the neo-behaviorist theories.

According to Miller and Charles' *Contextual Hypothesis* (Miller & Charles, 1991), the connection between **meaning, use, and context** is not simply an empirical method for the description of the facts of language, but a principle governing the way word meanings are learnt and used.[6]

In this approach, the basic building blocks of the semantic memory are *contextual representations*, defined as a "cognitive representation of a word in some abstraction or generalization derived from the contexts that have been encountered. That is to say, **a word's contextual representation is not itself a linguistic context, but is an abstract cognitive structure that accumulates from encounters with the word in various (linguistic) contexts**." (Miller & Charles, 1991, p. 5)

As pointed out in Lenci (2008), the empirical methodology devised in the Structuralist framework (*Weak Distributional Hypothesis*) has become a full-fledged cognitive hypothesis about the form and the origin of semantic representations (*Strong Distributional Hypothesis*). Word distributions, from being the epiphenomenon of a somewhat mysterious entity (meaning), are now credited a specific causal role in the formation of the semantic representations.

In the early 1990s, context-based theories of word meaning met the progress made in Computational Linguistics, and, in particular, the development of the first DSMs for the needs of Natural Language Processing and Information Retrieval (Deerwester et al., 1990; Schütze, 1992), which were in turn inspired by the seminal work of Salton et al. (1975). Such DSMs represented an extremely promising tool for testing psychological hypotheses concerning words and concepts, because they were, in a sense, concrete implementations of feature-based theories of semantic representations.

In the late 1990s two DSMs were developed for the purposes of cognitive modeling, that are still considered the prototype of the respective model classes: Hyperspace Analogue to Language (Lund & Burgess, 1996) and Latent Semantic Analysis (Landauer & Dumais, 1997). Given the historical/introductory nature of this section, we will not provide technical details concerning the implementations of those models (see Section 2.3), and we will instead discuss their status as cognitive models and their contribution to the development of the Distributional Hypothesis.

Hyperspace Analogue to Language (henceforth, HAL) is a term-based model developed by Lund & Burgess (1996) to provide the basis for a representational model of semantic memory. Co-occurrence information between words is collected from a corpus, resulting in vectors of the type $word = (w_1 = x, w_2 = y, w_3 = z, w_4 = r)$; co-occurrence frequencies are collected keeping track of the position of the feature word with respect to the target (i.e., storing separate co-occurrence counts for context words occurring to the left and to the right of the target) and of the distance between them (e.g., given the sequence "*the horse fell*" with *fell* being the target, *horse* is assigned a higher frequency value than *the*). Burgess and Lund suggest that HAL dimensions can be seen as a corpus-based version of the hand-coded lists of concept features (or semantic norms) traditionally used in psychology, but of a larger scale and of a better quality: corpus-

[6] "What people know when they know a word is not how to recite its dictionary definition they know how to use it (when to produce it and how to understand it) in everyday discourse [...]. Knowing how to use words is a basic component of knowing a language, and how that component is acquired is a central question for linguists and cognitive psychologists alike. The search for an answer can begin with the cogent assumption that people learn how to use words by observing how words are used. And because words are used together in phrases and sentences, this starting assumption directs attention immediately to the importance of context." (Miller & Charles, 1991, p. 4)

based feature representations based on statistical distributions ensure that the resulting semantic representation will not be dependent on pre-defined list of features (e.g., ANIMATE, MAMMAL, etc.); moreover, DSM semantic representations are acquired "in an unsupervised fashion in a noisy, conversation-like environment" (Lund & Burgess, 1996, p. 207): that is to say, the setting in which corpus-based features are collected is more natural and representative than a psychology laboratory.

Latent Semantic Analysis (henceforth, LSA) is a document-based DSM developed by Landauer & Dumais (1997) to address Plato's induction problem, the "mystery of excessive learning": "the fact that people have much more knowledge than appears to be present in the information to which they have been exposed". While HAL dimensions correspond to features of the target concepts, the ones in LSA correspond to possible topics of the documents in which the target word occurs. The implementation of LSA follows two steps: first, frequency information concerning the occurrence of target words in a collection of documents is collected, resulting in vectors of the type $word = (d_1 = x, d_2 = y, d_3 = z)$; then, LSA semantic representations are abstracted from the raw occurrence information through an inductive generalization process implemented with a mathematical data analysis technique: Singular Value Decomposition (henceforth, SVD). SVD captures latent similarities between model dimensions, producing a space with reduced dimensionality. For more details on SVD, refer to section 2.3.3.2.

LSA and HAL were successfully tested for the computational simulation of tasks connected to the representation, acquisition and processing of semantic knowledge. The assumption behind those simulations was that, if the representations encoded in the DSMs were comparable to the speakers' semantic representations, then the computational models would have exhibited similar performance patterns to the human subjects in the modeled experiments. LSA reached human-level performance in the TOEFL multiple choice semantic similarity task and it proved capable of acquiring knowledge of the full English vocabulary at a comparable rate to school children; HAL was successfully tested on the simulation of semantic priming effects (Lund & Burgess, 1996), on other tasks related to semantic/syntactic knowledge (Burgess & Lund, 1995) and as the basis for a model of cerebral asymmetries in lexical/semantic processing (Burgess & Lund, 1998).

2.1.3 The linguist's view on DSMs

Section 2.1.2 reviewed the success of early DSMs in the simulation of human semantic competence. Since then, DSMs have been successfully used to tackle a wide variety of further tasks in Natural Language Processing (D. Lin, 1998), Cognitive Modeling (Padó & Lapata, 2007; T. Mitchell et al., 2008; Murphy et al., 2012; Bullinaria & Levy, 2013), and in supporting theoretical linguistics with the modeling of thematic fit and verb classes alternations (Baroni & Lenci, 2010). Chapter 3 provides an extensive overview of all the tasks in which DSMs have been evaluated. Despite their success in practical applications and in language modeling, DSMs have been questioned both as linguistic and as cognitive models of semantic knowledge.

In this section, we review the challenges coming from Theoretical Linguistics, and we show how (and to what extent) the issues concerning the adequacy of the semantic representation encoded in DSMs have been addressed theoretically and have been solved by devising appropriate applications to exploit the information encoded in DSMs lexical entries.

2.1.3.1 DSM representations and word polysemy

Early criticism to DSMs targeted their **inadequacy in the representation of the meaning of polysemous words**. As each target word is represented by a unique aggregated distributional vector extracted from all its contexts, the senses of polysemous words get conflated, sort of flattened into the same representation. DSMs, however, proved successful in word sense disambiguation, the task of discriminating among the senses of a target word on the basis of the distributional information provided by the context of use. In what follows, we sketch pieces of work which are representative of two approaches to the task: disambiguation based on raw co-occurrence in the seminal work by Schütze (1998), and syntax-aware disambiguation (Erk & Padó, 2008; Thater et al., 2011).

Schütze (1998) showed how word-based DSM can be used to identify word senses in a two step process. First, distributional lexical entries for all words in the corpus are collected, with a bag-of-words approach gathering DSM vectors for word types: at this stage, a polysemous word like *suit* (which is ambiguous between the legal and the piece of clothing sense) is indeed assigned a unique, unsatisfactory vector. In a second step, the disambiguation algorithm goes through the corpus and builds bag-of-words representations for all the sentences in which potentially ambiguous words occur. Bag-of-words representations for sentences are calculated as the average (centroid) of the vectors of the words occurring in the sentence. Under the assumption that polysemous words are used only with one sense per sentence, bag-of-word sentence vectors for polysemous words are then clustered (i.e., grouped according to their similarity), identifying the different senses for a target word.

Erk & Padó (2008) propose a *Structured Vector Space* model for word meaning in context which integrates the notion of selectional preference in the process of disambiguation. To build context-adjusted representation of target words, for example *draw* in *draw a horse* vs. *a horse draws*, Erk & Padó (2008) combine the lexical vector of the target *draw* with the vector of *things that are typically done to a horse* (inverse object preference of horse) and with the vector of *things that a horse typically does* (inverse subject preference of horse). From a theoretical point of view, the argument-based disambiguation implemented here corresponds to the co-composition process described in Pustejovsky (1998): it is a type functional application in which the predicate selects the argument, and the argument picks up a meaning component of the predicate.

2.1.3.2 Semantic similarity in DSMs is too underspecified

One of the main criticisms against DSMs is that their implementation of **semantic similarity may be just too broad to be useful**, as it encompasses a wide range of relations with different logical properties (Sahlgren, 2006; Padó & Lapata, 2007; Lenci, 2008). Much research has been conducted in the direction of a refinement of the distributional notion of semantic similarity, showing that DSMs can learn similarity among relations between pairs of words (Turney, 2008; Baroni & Lenci, 2010), and that they can also learn relation-specific representations to discriminate between pairs of candidate relations: synonymy vs. antonymy (Scheible et al., 2013; Santus et al., 2014), hypernymy vs. hyponymy (Weeds & Weir, 2003; Weeds et al., 2004; Lenci & Benotto, 2012).

One of the strategies adopted for the classification of synonym/antonym pairs (e.g.,

happy/sad) is to identify the appropriate contextual features to support the distinction between competing relations: Scheible et al. (2013), to mention just one example, found that verbs are better than nouns when used as distinctive features between pairs of antonymous adjectives (*happy people smile* vs. *sad people cry*). Another strategy mentioned in the literature on this task is to quantify the extent and the salience of the intersection between the two words, under the assumption that synonyms share a significantly higher number of contexts than antonyms (Santus et al., 2014).

The classification of hypernymy/hyponymy relations has received plenty of attention in Distributional Semantics. The main DSM approaches to this issue are based on the *Distributional Inclusion Hypothesis* (Geffet & Dagan, 2005; Kotlerman et al., 2010), which is the intuition that for a given pair of hyponym/hypernym words, the features of the hyponym should be a subset of the features of the hypernym (Weeds et al., 2004; Clarke, 2009; Lenci & Benotto, 2012).

2.1.3.3 DSMs and reference

From large amounts of generic and episodic mentions of words collected from a corpus, DSMs abstract a generalization which has to do with world-knowledge more than with linguistic categories. As frequently remarked in the literature and summarized in Baroni, Bernardi, & Zamparelli (2014), DSMs ("if they are able to extract any factual information at all") can capture **generic knowledge**, a typical application of corpus-based semantics being "the extraction of commonsense-knowledge factoids that are generally useful while not universally true: bananas are yellow, birds fly, etc." (p. 258).[7]

DSMs represent the meaning of symbols (words) as distributions over other symbols (words), and are therefore are unable to address those aspects of word meaning that have to do with **reference** to the world. The issue of reference in DSMs has been addressed both theoretically (Baroni, Bernardi, & Zamparelli, 2014) and experimentally (Herbelot, 2015), but is far from being fully solved.

Baroni, Bernardi, & Zamparelli (2014) point out that "[...] the divide between DSM and denotational semantics is not reference/lack of reference, but rather reference to *linguistic strings* (which we can easily record) or to *objects* (which we cannot)." (p. 259). Distributional Semantics stands in a complementary relation to denotational semantics, and the authors suggest that when we hear a sentence, the distributional representation of its constituents helps us to build a "sketch of the typical contexts in which it can be uttered truthfully, which can orient our perceptual system to pick up the relevant cues to determine if a dog is indeed barking right now, so that we can evaluate the referential meaning of the sentence." (p. 260).

[7] Whether corpus-based models can provide an exhaustive representation of the set of features associated to concepts has been (and still is) a matter of debate in the psychological and computational linguistics literature (Murphy, 2002; Baroni & Lenci, 2008; Baroni et al., 2010). Corpus-based models proved to be suited to capture properties related to taxonomic (e.g., "birds – animals"), or script knowledge (e.g., "birds – sky") and functional properties (e.g., typical actions such as "birds – fly"), with large variation due to the different model implementations (see Baroni & Lenci (2008) for an overview). The extraction of visual features (e.g., "bananas – yellow") from co-occurrence information remains, however, problematic, because typical visual properties are often not among the most frequent collocates of a target word; recent work showed how the integration of visual information extracted from images in the corpus-based representation encoded in "standard" distributional vectors can significantly improve DSM performance with respect to the identification of typical visual features (Bruni et al., 2012).

Herbelot (2015) reports experimental work on a subclass of referring expressions with well established theoretical properties: proper names. It is shown that DSMs can successfully (i.e., with good performances, and in a theoretically sound way) model the meaning of proper names of fictional characters. More specifically, proper names are encoded in context-adjusted representations of *kind* vectors: "We propose that on encountering Mr Darcy for the first time, a reader might simply attribute him the properties of the lexical item *man*, as given by the relevant distribution in a large corpus, and then specialize the representation as per the context where Darcy occurs."(Herbelot, 2015, p. 158) First, the bag-of-words vector for the *individual Mr Darcy* is extracted from a novel, which is additionally annotated with proper name semantic classes. As *Mr Darcy* is labelled as *man*, the distributional vector of the lexical item *man* (extracted from larger reference corpus) is selected as *kind* vector. The *MrDarcy-instance-of-man* vector is obtained by contextualizing the *man* vector with respect to the contexts in which *Mr Darcy* occurs; in practice, this is achieved by re-weighting the components of *man*. The re-weighting algorithm has specific parameters which guarantee that instantiation (the property of being an instance of the relevant kind) does not overwrite uniqueness (the property of being a unique entity) and individuality (the property of being separable from the kind vector) of the proper name.

As a consequence of the lack of appropriate DSM representations for function words, an account for the meaning of other classes of referring expressions (such as definite descriptions) has not yet been developed.

2.1.3.4 DSMs and entailment

Entailment ($\models$) is a fundamental logical property of natural language. An entailment relation holds between a proposition (antecedent) and another proposition (consequent) when the truth of the first implies the truth of the second, in any circumstance. Even if in Formal Semantics truth values can only be assigned to propositions (sentences), entailment relations are also established between smaller linguistic units, such as words (*car* $\models$ *vehicle*) or phrases (*all cars* $\models$ *some cars*).

Section 2.1.3.2 discussed the DSM approaches for modeling the hypernymy relation (*car* $\models$ *vehicle*), an instance of *lexical entailment*. Knowing that *car* is a hyponym of *vehicle* allows humans and machines to recognize that if *John drives a car* is true, *John drives a vehicle* is true as well.

Besides the lexical level, entailment relations also hold between quantified phrases (*all cars* $\models$ *some cars*). From a set-theoretic point of view, quantifiers express relations between sets (e.g, the set of the entities that are cars, the set of the entities that are polluting). Knowing that a property which applies to *many* members of a set also applies to *some* members of the same set allows to draw textual inferences such *many cars are polluting* $\models$ *some cars are polluting*. Inferences drawn from explicitly quantified statements have been widely investigated in natural logics (MacCartney & Manning, 2008). Baroni et al. (2012) show that DSM representations can also be used to identify pairs of entailing quantifiers (e.g., many $\models$ some; many $\not\models$ all). DSMs vectors are extracted for quantified phrases (e.g., $\vec{v}_{many\ cars}$, $\vec{v}_{no\ cars}$, $\vec{v}_{all\ cars}$), and are fed in pairs to a Support Vector Machine classifier as positive (e.g., $\vec{v}_{many\ cars}$, $\vec{v}_{some\ cars}$) and negative (e.g., $\vec{v}_{many\ cars}$, $\vec{v}_{no\ cars}$) training examples. Baroni et al. (2012) show that the SVM classifier successfully learns to classify unseen pairs of quantified phrases as entailing or not entailing, on the basis of the information encoded in the DSM vectors,

without relying on explicit entailment rules.

2.1.3.5 DSMs and non-distributional semantic knowledge

In the previous section, we discussed how DMSs can be used to model entailment relations between quantifiers. Sentences containing quantified phrases (e.g., *all dogs are mammals*) can be interpreted as probabilistic statements concerning the relation between a target concept (*dog*) and one of its defining features (*is_a_mammal*). If a speaker utters the sentence *all dogs are mammals*, this means that, according to his representation of what the world is like, he is totally certain that every *dog* will have the property of being a *mammal*. Such probabilistic mapping is not always linguistically encoded by explicit quantifiers: properties can also be ascribed to concepts with the use of kind-denoting bare plurals (e.g, *birds have wings*). The interpretation of bare plurals is, however, often ambiguous (e.g, *birds have wings*=ALL, *birds fly*=MOST) and it is heavily dependent on the non-distributional knowledge speakers resort to while interpreting the linguistic input.

In this connection, Herbelot & Vecchi (2015) contribute to a set-theoretic characterization of the interface between distributional and non-distributional semantic knowledge by showing that DSMs can be used to learn weighted relations between concepts and their features. In their work, they assume "the existence of a mapping between language and a shared set of beliefs about the world, as negotiated by a group of speakers." (p. 23). Such a mapping can be learnt as a functional relation between two vector spaces. The first vector space is a standard bag-of-words model. The second vector space, corresponding to the "shared set of beliefs about the world" and defined "set-theoretic", is based on a dataset of feature norms (McRae, Cree, et al., 2005): context dimensions correspond to features (e.g., for the target *bird*: *has_wings*, *flies*, *sings*), and the target-context values are manually annotated with general quantifiers (e.g, *bird – has_wings* = ALL; *bird – flies* = MOST) converted to numerical values (e.g., ALL=1,SOME=0.35, etc.). A function to map distributional vectors into feature vectors is learnt with linear regression, exploiting the systematic correspondences between context dimensions in the two spaces. The mapping function is then used to produce weighted feature-based representations by taking DSM vectors as input (e.g., for *cat*, predict 1 (ALL) for *mammal*, 0 (NO) for *human*, etc.). The learnt mapping space replicates with high correlation the quantifier information from the gold annotation, and it also proved successful in the generation of quantifiers for target/feature pairs from the test data.

2.1.3.6 DSM representations above the word level

The criticisms reviewed so far are either concerned with the limitations of DSM lexical representations for single words (polysemy, reference, concept features) or pertain to the comparison between word vectors (semantic relations, entailment at the lexical level). An adequate model of lexical meaning should, however, also allow the construction of meaning representations for complex expressions, based on the meaning of its components. The notion of **compositionality** (the meaning of sentences is built incrementally by combining the constituent meanings) is crucial in Formal Semantics, and it has been extensively addressed in Distributional Semantics in the last years.

Broadly speaking, the strategies for implementing compositionality in Distributional Semantics fall into two categories.

The first strategy is **vector mixture** (J. Mitchell & Lapata, 2008, 2010). Given the distributional vectors for two words, $\vec{a}$ and $\vec{b}$, a composed vector $\vec{c}$ is calculated as a mathematical combination of $\vec{a}$ and $\vec{b}$. The most common methods for vector combination are the addition and multiplication of corresponding vector components: as both operations are symmetric, they are insensitive to word order and to potential predicate-argument asymmetries holding between $\vec{a}$ and $\vec{b}$.[8] Additive methods for vector composition were the most common in early work in Distributional Semantics (Landauer & Dumais, 1997; Schütze, 1998): in this case, the composed vector $\vec{c}$ retains all the features (components) in $\vec{a}$ and $\vec{b}$ (union of $\vec{a}$ and $\vec{b}$); in the case of vector element-wise multiplication, $\vec{c}$ will retain only the features shared by $\vec{a}$ and $\vec{b}$ (intersection of $\vec{a}$ and $\vec{b}$). J. Mitchell & Lapata (2010) compared additive and multiplicative methods in the task of predicting similarity judgments (i.e., how similar are *little dog* and *big cat*?). Multiplication turned out to be the best-performing method, and it proved to be the most competitive also in further work on composition tasks (E. Grefenstette & Sadrzadeh, 2011; Vecchi et al., 2011; Boleda et al., 2012)

The second approach to composition in DSM is a syntax-driven one in which composition in the distributional space is implemented as **function application** (Baroni, Bernardi, & Zamparelli, 2014). Building on Frege's (1892) distinction between *complete* and *incomplete* expressions, Baroni, Bernardi, & Zamparelli (2014) distinguish between classes of words whose meaning can be defined in terms of a distributional vector (such as nouns) and words whose meaning is better described as the transformation undergone by the vectors of the words they modify (such as adjectives and verbs). Distributional vectors for nouns are extracted in the standard, bag-of-words fashion, while verbs and adjectives are learnt as distributional functions. Composition is implemented by applying distributional functions (e.g., f_{small} for the adjective *small*) to lexical distributional vectors (e.g., $\vec{v}_{cat}$) or to other functions, yielding distributional representations for expressions of potentially arbitrary complexity. In the case of adjective-noun composition, the output of the function application is a predicted vector which approximates the vector of the target composed expression: $f_{small}(\vec{v}_{cat}) \approx \vec{v}_{small\ cat}$.

Distributional functions for verbs and adjectives are learnt in a two-step process. Let us suppose we want to learn the distributional function for the adjective *small*: f_{small}. In the first step, the distributional vectors for all nouns in the corpus are collected ($\vec{v}_{dog}$, $\vec{v}_{house}$, etc.), as well as the vectors for all adjective-noun combinations in which *small* acts as a noun modifier ($\vec{v}_{small\ dog}$, $\vec{v}_{small\ house}$, etc.), in the same feature space. In the second step, **linear regression** is used to learn the distributional function f_{small} through a comparison between its input and the desired output: $f_{small}(\vec{v}_{dog}) = \vec{v}_{small\ dog}$, $f_{small}(\vec{v}_{house}) = \vec{v}_{small\ house}$, etc. The distributional function f_{small} has the form of a matrix of weights describing how the position of nouns in the semantic space changes when the adjective *small* modifies them. Thanks to its mathematical properties, the regression learner is able to capture complex displacement dynamics which go beyond vector shifting (summation), reweighting (element-wise multiplication), and linear scaling of feature dimensions.

This approach proved successful in a number of tasks (see Baroni, Bernardi, & Zamparelli (2014), and references therein), including the assessment of the similarity of complex expressions ($\vec{v}_{little\ dog}$ vs. $\vec{v}_{little\ criminal}$) and the detection of semantically

[8] Given the non-technical nature of this section, we do not discuss here *weighted* additive and multiplicative methods.

anomalous expressions (i.e., finding out that, out of two equally unattested adjective-noun combinations, such $\vec{v}_{coastal\ mosquito}$ vs. $\vec{v}_{coastal\ subtitles}$, the latter is a less plausible combination).

2.1.4 The grounding problem

In the previous section a number of criticisms were reviewed, targeted at DSMs as linguistic models of semantic knowledge. More than criticisms, those issues can be considered as "desiderata" for a theoretically adequate representation of meaning. We showed that a large portion of them has been successfully addressed by devising the appropriate DSM application. In this section, we turn to criticism from Psychology and Cognitive Science, and to the **grounding problem**: DSMs which are built only from textual data, are disembodied from the world and they turn out to be inadequate for the representation of human semantic knowledge. We show that this problem cannot be addressed DSM-internally (language corpora are not enough), but it requires the integration of multi-modal information in distributional representations

Glenberg & Robertson (2000) argue that DSMs cannot be considered psychologically valid models of semantic representations, because they define abstract symbols (words) in terms of their relations with other abstract symbols (words) and because symbol co-variation cannot, be considered an adequate representation of meaning by itself (Searle, 1980). In this connection, the need for grounded semantic representations (i.e., representation anchored in the real world) can be considered the psychological/neuroscientific counterpart of the reference problem which has been discussed in the previous section from a theoretical linguistics point of view.

Embodied theories of cognition (Barsalou, 1999), with robust support from neuroscientific data (see Barsalou (2008) and references therein), establish that concepts are inherently modal entites. Concepts (and, as a consequence, word meanings) are acquired *and stored* in the sensory-motor system: the meaning of a word is an embodied simulation which allows the speaker to re-enact the perceptual experiences associated with the corresponding concept.

The approaches to the grounding problem in Distributional Semantics are characterized by a shared global strategy: bringing together perception and distribution by learning a mapping between perceptual and distributional data. They differ, however, with respect to the way perceptual data are represented. One possibility is to rely on speaker generated features as a rough approximation of perceptual data (Andrews et al., 2009).[9] As an alternative, actual visual information can be integrated with distributional information. Bruni et al. (2013) propose an implementation of this "radical" approach to the integration of perceptual information in DSMs: by exploiting computer vision techniques, visual information is extracted from images labeled with words. Bag-of-visual-words are extracted, which extend the bag-of-words DSM concept to images, "describing them as a collection of discrete regions, capturing their appearance and ignoring their spatial structure (the visual equivalent of ignoring word order in text)." (Bruni et al., 2013, p. 7). Bag-of-words and bag-of-visual-words are integrated in a

[9]The approach described in Herbelot & Vecchi (2015) is definitely an instance of integration of distributional (a bag-of-words model) and non distributional (a vector space based on feature norms) information. Given its focus on the language/logic interface, however, we considered it less representative of DSM approaches to the grounding problem than of the application of DSMs to tackle theoretical linguistics issues.

multimodal distributional semantic model which proved superior to a purely text-based approach, in a number of empirical tests.

Vinson et al. (2013) point out that the grounding problem, initially perceived as a major limitation of the Distributional Hypothesis (and one that was impossible to overcome), in fact led to a redefinition and extension of the hypothesis: "Rather than the relevant context of a word being just its linguistic context, we can extend the definition of context to include the extralinguistic or real-world context in which the word occurs, 'combined models' in other words. From this, the more general distributional hypothesis is that the meaning of a word is acquired from the contexts of its usage, regardless of whether these contexts are intralinguistic or extralinguistic." (Vinson et al., 2013, p. 141)

2.2 Formal definitions of DSMs

As anticipated in the introductory section, the main goal of this chapter is to provide a taxonomy for the multiple design choices that are available when constructing and using a DSM. As pointed out in Lowe (2001), a fundamental preliminary step towards a proper understanding of DSMs is an explicit mathematical formulation of the overarching theoretical framework in which DSMs are grounded. In this section, we review formal definitions proposed in the literature (Lowe, 2001; Padó & Lapata, 2007) and discuss their limitations and possible extensions to account for different classes of DSMs.

Lowe (2001, p. 676) defines a a semantic space model as a "method of assigning each word in a language to a point in a real finite dimensional vector space". Each target word t is assigned a **distributional profile**, which has the form of a vector containing frequency information concerning the occurrence of t in a set of documents or its co-occurrence with other words in the corpus. In Lowe's account, a semantic space is formally defined as a quadruple $\langle \mathrm{B}, \mathrm{A}, \mathrm{S}, \mathrm{M} \rangle$:

- A set of basis elements (B) which are considered to be representative contexts and constitute the dimensions of the semantic space (inflected words, lemmas, documents);
- A lexical association function (A), which turns co-occurrence frequencies between target words and basis elements into values which are assigned to the corresponding positions of the distributional profiles. As the application of a lexical association function is optional, assigned values can also be identical to co-occurrence frequency; given the properties of word distributions, though, it is recommended to use association measures which provide a more sensible interpretation of co-occurrence frequency, and account for frequency bias (Evert, 2008);
- A similarity measure (S) which maps pairs of vectors into a continuous value which quantifies their contextual similarity;
- A transformation (M), optionally applied to map a semantic space (the DSM matrix) into another semantic space, typically with fewer (and, at least theoretically, more meaningful) basis elements (i.e., fewer dimensions).

A limitation of the formal definition proposed by Lowe (2001) concerns the status of the co-occurrence extraction criteria (e.g., size of the context window), which cannot

be framed in any of the defining elements of the proposed quadruple: B only defines the valid contexts, A applies to already extracted co-occurrences, S and M are concerned with further steps in the manipulation and use of the distributional information encoded in the semantic space matrix. The extraction of co-occurrences from a source corpus is a crucial step in the construction of a DSM and it involves a number of design choices which will be thoroughly reviewed as DSM parameters in sections 2.3.1 to 2.3.2.3. We summarize them here to better define the scope of the phenomena a formal definition for DSMs needs to account for.

As pointed out in Evert (2008) the operationalization of the notion of co-occurrence requires a precise definition for the "nearness" of two words.[10] If a *surface-based* view on co-occurrence is adopted, two words are said to co-occur if they appear close to each other within a distance which is quantified in terms of intervening words; such distance, defined *context window* or *collocational span*, is set as an extraction parameter; further design choices, just to mention a few, are: whether to use both the left and the right context; whether to stop at sentence boundary while extracting co-occurrences; whether to consider punctuation when computing the context window. Table 2.1 illustrates the extraction of co-occurrence information from the sentence *"a cute dog barks"*, with a context window of size 3. The extraction window slides through the text collecting co-occurrence counts for targets/nodes based on the absolute difference between the positional index of the target, $ind(t)$, and the positional index of the collocate.

ind(t)-3	**ind(t)-2**	**ind(t)-1**	**t**	**ind(t)+1**	**ind(t)+2**	**ind(t)+3**
#	#	#	**a**	cute	dog	barks
#	#	a	**cute**	dog	barks	#
#	a	cute	**dog**	barks	#	#
a	cute	dog	**barks**	#	#	#

Table 2.1: *A cute dog barks* - surface co-occurrence

If a *syntax-based* perspective to co-occurrence is adopted, "nearness" is defined by the presence of a direct syntactic relation between the node and a collocate (e.g., direct object, subject, etc.); in our running example, if we focus on the subject relation the only node-collocate pair extracted would be <*barks, dog*>.[11]

Padó & Lapata (2007) propose an extension of the formal definition for DSMs devised by Lowe (2001). Padó & Lapata (2007) explicitly design their formal definition for an application to syntax-based DSMs, but they point out that it could be also be applied to surface-based DSMs. We adopt their formal definition and spell out the assumptions, formal details, and necessary extensions for its application to the parameter space of surface-based DSMs.

The basic building block of the formal definition proposed by Padó & Lapata (2007) is the notion of *path* (π). A *path* is a **co-occurrence pattern connecting a target**

[10] Following the Corpus Linguistic terminology, we use the term *node* to refer to the word whose co-occurrence profile we are interested in (corresponding to the *target* in the DSM terminology) and *collocate* to refer to a word occurring "near" the node (roughly corresponding to the notion of *feature* in a DSM).

[11] A further option is *textual co-occurrence*, according to which two words co-occur if they appear in same textual unit (e.g., a sentence, a paragraph, a document). If the adopted textual unit is a sentence, this approach is equivalent to a surface-based one with no pre-established window size.

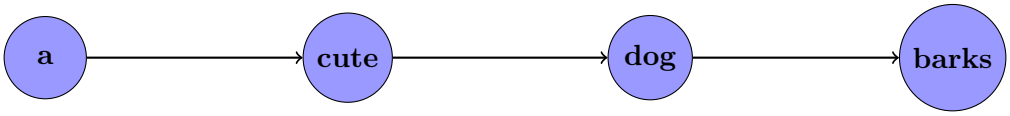

Figure 2.1: *A cute dog barks*: precedence graph

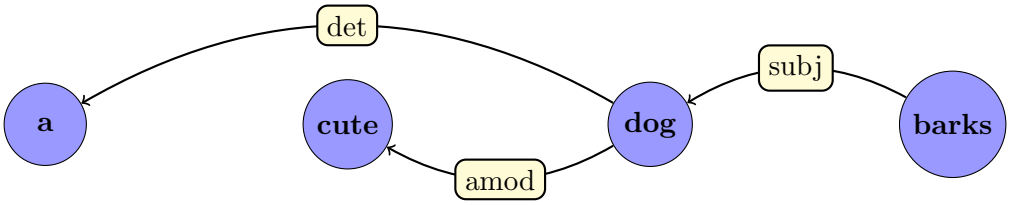

Figure 2.2: *A cute dog barks*: syntax-based graph, labelled with dependency information

word with a context word. The notion of co-occurrence path relies on a graph-based representation of the textual unit; here, we adopt the sentence as a textual unit within which co-occurrence is computed. Graph nodes correspond to inflected words, optionally annotated with lemma and part of speech information. The edges of the graph can encode different types of information: simple linear order (from the first word in the sentence to the second one, to the third, and so on) or syntactic relations holding between the nodes (from predicate words to their arguments, e.g., from a verb to its subject): window-based DSMs rely on the former, dependency-based DSMs on the latter. Note that the standard implementation of surface-based co-occurrence (table 2.1) can be derived from a graph representation of linear ordering (figure 2.1).

The graph representation for the sentence *a cute dog barks* is displayed in figures 2.1 (precedence graph encoding linear order information) and 2.2 (syntactically structured version). The comparison between the two graphs highlights a further (optional) property of graph edges: their **label**. Edge labels characterize the nature of the relation holding between the connected nodes: in the graph in figure 2.2, we identify a *subject* (label: subj) relation between *barks* and *dog*, an *adjectival modification* relation (label: adj) between *dog* and *cute*, and a *determiner* relation (label: det) between *dog* and *a*. In surface-based graphs, the **direction** of the edges mirrors the linear order of the words in the text, while in the syntax-based graphs it encodes dependency information (i.e., the edge points at the argument).

The graph representations in 2.1 and 2.2 are quite restrictive with respect to the available paths. In the graph in 2.1, for example, only the right context is accessible and no co-occurrence information is available for *barks*. In the updated graph in 2.3 the full context is accessible for each word, thanks to the introduction of inverse edges pointing in the opposite direction and labelled accordingly (right vs. left). In a similar

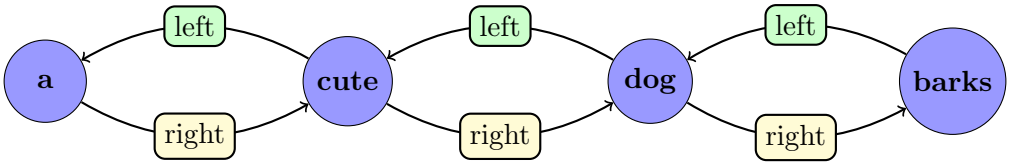

Figure 2.3: *A cute dog barks*: surface-based graph, labelled with direction information

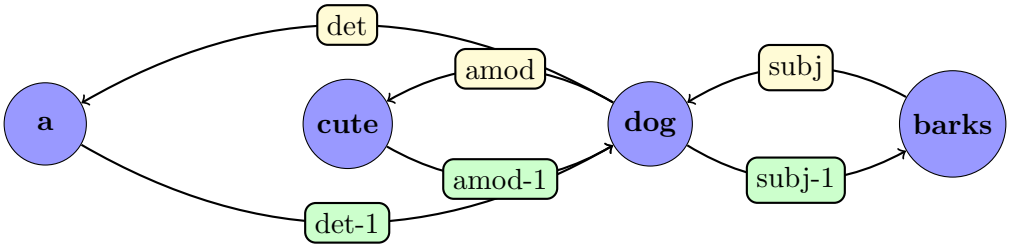

Figure 2.4: *A cute dog barks*: syntax-based graph, with inverse dependencies

way, in 2.2 we can reach *dog* starting from *barks*, but not the other way round. Figure 2.4 shows an updated version of the original graph, in which inverse dependencies (Erk et al., 2010) are introduced and labelled accordingly (a "-1" string is concatenated to the relation label).

Given the graph representation for a sentence s, a *path* π is defined as an ordered sequence of nodes **anchored** (i.e., starting) at a source node a and ending at a node b. To be efficiently employed in the construction of a semantic space, paths need to be **acyclic**, i.e., each node can be traversed only once in the same path. A number of basic operations can be devised, which apply to a path and return its starting node ($start(\pi)$), its end node ($end(\pi)$), and its length (equal to the number of its edges). If edge labels are available, the labels of the edges composing a path can be concatenated to assign composite labels to a path; a labelling function ($label : \Pi \rightarrow R$) is defined, which maps paths (Π) into sequences of edge labels (R).

In Padó & Lapata (2007) account, a semantic space is formally defined by the tuple $\langle \mathbf{T}, \mathbf{B}, \mathbf{M}, \mathbf{A}, cont, \mu, \nu \rangle$.[12] The extension proposed in this thesis builds on their formal definition and introduces two further mapping functions: μ_b and μ_t. In more detail, the updated formal definition contains:

- A set of target elements T, a set of basis elements B, and a co-occurrence matrix $M = T \times B$.

- A **context selection function**: $cont(t) = \{\pi \in \Pi_t | \ criteria(\pi)\}$
  Given a target word (t) the context selection function determines which paths of all those anchored at it (Π_t) can be considered distributionally informative on the basis of one or more *criteria*. Context selection can operate on path length: for example, the function $cont(t) = \{\pi \in \Pi_t \mid \|\pi\| = 1\}$ selects only paths linking the target to immediately adjacent words in the sentence graph. For an extensive review of the context selection criteria, see section 2.3.2.

- A **path value function** $\nu : \Pi \rightarrow \mathbb{R}$
  The path value function specifies the relative contribution of different paths to the quantification of co-occurrence. The basic option is to assume that all paths have the same weight: $\nu_{plain} = 1$. This approach characterizes the mainstream take on window-based co-occurrence, according to which all words in the context window

[12] Note that while Lowe (2001) uses M for matrix transformation, Padó & Lapata (2007) use this symbol to refer to the co-occurrence matrix. Matrix transformation (Lowe's M) does not enter the formal definition by Padó & Lapata (2007) because their experiments do not involve dimensionality reduction, but it can be unproblematically introduced as an additional element.

(e.g., 2 words to the left and to the right of the context) have the same weight in the distributional profile for a target word t. Refer to section 2.3.2.2 for more details concerning the path value function.

- Three **mapping** functions:
  - A **basis mapping function**: $\mu : \Pi \rightarrow B$
    Given a target word (t) and the subset of paths anchored at t defined by the context selection function $cont(t)$, the basis mapping function creates the dimensions of the semantic space by collapsing paths that are considered equivalent. The most elementary basis mapping option deems all paths ending at the same word equivalent: $\mu(\pi) = end(\pi)$; this take on basis mapping characterizes, for example, bag-of-words models with an undirected context window. More examples are discussed in section 2.3.2.3.
  - A **target mapping function**: $\mu_t : T \rightarrow T$
    The target mapping function is introduced here as an extension of Padó and Lapata's formal definition, to make the framework fully flexible in terms of DSM parametrization. If node annotation (e.g., stem, lemma, and/or part of speech) is available, the target mapping function μ_t maps the target to one of its available versions; for example, it maps targets from their inflected forms (e.g., *books*) to part-of-speech disambiguated lemmas (*book_verb*). See section 2.3.2.3 for further discussion of target mapping properties.
  - A **secondary basis mapping function**: $\mu_b : B \rightarrow B$
    The secondary basis mapping function is introduced here as an extension of Padó and Lapata's formal definition, to account for pattern-based co-occurrence (Baroni & Lenci, 2010). Secondary basis mapping operates on the dimensions of the semantic space after a "first pass" of basis mapping and frequency aggregation; it is employed when paths have a complex structure, like in Baroni and Lenci's *Distributional Memory* and operates on the path component of the basis element which is usually reduced to a "more general" subpart, leaving the context word unaffected. For more details on the approaches for the computation of co-occurrences (frequency aggregation vs. pattern-based co-occurrence), see section 2.3.2.4.
- A lexical association function A and a similarity measure S, whose definition fully correspond to the one proposed by Lowe (2001);

The crucial extension of the formal definition by Padó & Lapata (2007) with respect to the one by Lowe (2001) is represented by the three functions which regulate the extraction of co-occurrence information: the context selection function, the basis mapping function, and the path value function. In particular, the separate treatment of co-occurrence paths and basis elements makes the framework flexible and suitable as a general framework for vector based models. In this section, we discussed how it can be made fully compatible with window-based DSMs if a graph representation of surface co-occurrence is assumed; the formal definition defined in this section is complemented by the discussion of strategies of frequency quantification (frequency aggregation vs. pattern-based frequency) discussed in section 2.3.2.4.

The formal definition by Padó & Lapata (2007) is explicitly **designed for term-based DSMs**. Document-based DSMs could be accounted for by representing both

documents and words as nodes, with edges encoding the occurrence of a target word t in a document d, corresponding to a basis element; the generalization power of the extraction functions would, however, stay unexploited, as a word either occurs in a document or doesn't. As already discussed in the introductory chapter, document-based DSMs fall out of the scope of this thesis.

2.3 DSM parameters: a taxonomy

The aim of this section is to provide an overview of the many design choices available when building a DSM. The focus of this chapter is therefore on DSM parameters, and not on DSM performance and evaluation which are discussed in chapter 3. It is, however, quite difficult to discuss DSM parameters without giving the reader at least a general idea of their effect on DSM performance: for this reason, in the remainder of this chapter, we will at times refer to DSM performance in standard semantic similarity tasks, without giving further details on the specific tasks. For reasons of space, the literature review conducted in this section will tackle DSM modeling conducted on English, and its scope of the literature review will be kept on medium-to-large scaled DSM evaluation studies.

This section is structured according to the steps to be taken to build a DSM (pipeline) and to the coordinates provided by the formal definition introduced in the previous section. We start from a preliminary step, namely **corpus selection and preparation** (section 2.3.1). We then proceed to the **extraction of co-occurrence information** (section 2.3.2), which we characterize in terms of the extraction functions defined by Padó & Lapata (2007). The output of the extraction step produces a co-occurrence matrix which already corresponds to a DSM, and can be used to compute similarity among target vectors; the literature on DSMs shows, however, that further **manipulation of the co-occurrence matrix** is almost always applied to improve the semantic representation: an overview of such matrix manipulation options, ranging from feature selection, weighting and thinning to dimensionality reduction is provided in section 2.3.3. Section 2.3.4 discusses the options available for the computation of semantic similarity.

2.3.1 Corpus selection and pre-processing

When building a DSM, the first choice to be made concerns the **corpus** from which co-occurrence information is extracted. The possible reasons for selecting a source corpus are theoretical (e.g., for cognitive modeling purposes, choosing a corpus which is representative of a speaker's language experience), practical (e.g, choosing a corpus whose size is manageable for the available computing resources, or for which linguistic annotation is already available), or simply determined by the state-of-the-art of the task at issue.

Corpora differ with respect to many features:

- Size: small corpora vs. large web corpora;
- Language type: written vs. spoken;
- Quality: professionally edited texts (book corpora) vs. web pages collected from the Internet without any further editing;

- Distribution of genres: focus on specific genres (e.g., narrative or encyclopedia/newspaper articles) vs. balanced sample of different genres;

- Register and degree of objectivity: colloquial and opinion-oriented (e.g., social media) vs. formal and factual (e.g., encyclopedia articles) vs. narrative and fictional (e.g., book corpora).

The literature on DSM evaluation reveals a preference for the following corpora, which are employed separately (one corpus, one DSM) or in a concatenation (three corpora, one DSM):

- The **British National Corpus**[13] (henceforth, BNC), a high quality 100 million token collection of written and spoken English texts from a wide range of sources. Evaluation work on the BNC has been conducted by Bullinaria & Levy (2007); Padó & Lapata (2007); Kiela & Clark (2014). BNC is also included as a subcorpus (e.g., concatenated with other corpora) in Distributional Memory by Baroni & Lenci (2010) and in the evaluation study by Baroni, Dinu, & Kruszewski (2014).

- Different snapshots of the **English Wikipedia**. Baroni & Lenci (2010) and Baroni, Dinu, & Kruszewski (2014) include the publicly available WaCkypedia_EN corpus,[14] a 2009 dump of the English Wikipedia (800 million tokens). Sridharan & Murphy (2012), Polajnar & Clark (2014) and Levy et al. (2015) employ a 2012 and a 2013 Wikipedia snapshot, both of 1.7 billions tokens Polajnar & Clark (2014). Kiela & Clark (2014) employ a sub-spaced version of Wikipedia (Stone et al., 2008), collected by querying it with the words in the evaluation datasets and collecting the top 10 ranked documents (10 million tokens). Despite the differences in size, the different dumps of Wikipedia share a high text quality due to peer rewiewing and constant updates, and a formal, fully factual register.

- The **ukWaC** corpus:[15] a 2 billion word corpus collected by crawling the Web within the .uk domain with medium frequency words from the BNC as seeds. Despite the cleaning procedure, language samples from ukWaC may present some of the typical features of web-crawled data, such as sentence repetitions, non-standard language, lists and residual html annotation. Due to its size and to the large coverage of topics, genres, and registers ukWaC is the most commonly employed corpus in large scale evaluation studies such as Bullinaria & Levy (2007, 2012); Kiela & Clark (2014); Baroni & Lenci (2010); Baroni, Dinu, & Kruszewski (2014).

Corpora of bigger sizes than BNC, Wikipedia and ukWaC are also available, e.g., EnCOW[16] (10 billions tokens) or the Google N-gram corpora: the Web Google N-grams, 1 trillion tokens (Brants & Franz, 2006), the Google Book N-grams corpus, 350 billions tokens in the English section (Michel et al., 2011), and the syntactically annotated version of the Google Book N-grams (Goldberg & Orwant, 2013). However, building, manipulating and evaluating a DSM from such large text collections is computationally

[13] `http://www.natcorp.ox.ac.uk/`
[14] http://wacky.sslmit.unibo.it/doku.php?id=corpora
[15] http://wacky.sslmit.unibo.it/doku.php?id=corpora
[16] http://corporafromtheweb.org/encow14/

very costly; it is therefore not common to conduct extensive evaluation on such corpora, with the exception of Sridharan & Murphy (2012) and Kiela & Clark (2014).[17]

The quality-quantity trade-off The relation between corpus size and corpus quality, and their joint impact on DSM performance have been investigated by Bullinaria & Levy (2007, 2012) and Sridharan & Murphy (2012).

Bullinaria & Levy (2007) compare BNC to a very poor quality corpus built from a random selection of Internet-based newsgroup messages (168 millions tokens) and Bullinaria & Levy (2012) compare BNC to ukWaC. Furthermore, Bullinaria & Levy (2007, 2012) build DSMs from different subcorpora of the selected source corpora (e.g., 10, 50, 100 millions tokens, etc.) and performances are compared on a number of standard semantic similarity tasks. This methodology allows the authors to draw empirical conclusions at two different levels: from a within-corpus point of view, they test the impact of quantity keeping quality constant; from a between-corpora point of view, they test whether the performance peak of smaller corpus is comparable to the performance of a larger corpus, at the same size (e.g., full BNC vs. 100 millions tokens from ukWaC). At full size, BNC outperforms the Internet newsgroup message corpus but is outperformed by ukWaC (though not dramatically in all tasks); at comparable sizes, however, BNC outperforms ukWaC (and it does so also for larger subcorpora of ukWaC), showing that larger amounts of textual material are needed to compensate for poor quality (Bullinaria & Levy, 2012, p. 896).

Comparable results are reported by Sridharan & Murphy (2012), who provide further experimental evidence for the strong impact of the corpus quality on DSM performance. Their study represents the only attempt to integrate the Google n-gram corpora in a large scale evaluation study; DSM evaluation is conducted on the following corpora (reported sizes are after preprocessing; see the original paper for more details): Google Web n-grams (353.4 billions tokens), Google books n-grams (199.4 billions), a corpus of Twitter texts (2.1 billions tokens), a 2012 dump of the English Wikipedia (1.7 billions). DSMs are evaluated in a neurolinguistic decoding task and in a number of standard semantic similarity tasks. Sridharan & Murphy (2012) show that a corpus of small size and high quality (Wikipedia) outperforms larger corpora of poorer quality at comparable corpus sizes (i.e., when the larger corpora are reduced to subsample of the same size of Wikipedia).

Linguistic annotation After selecting a source corpus, it is often necessary to perform a number of pre-processing operations on the raw text before extracting co-occurrence information, such as tokenization, normalization (case-folding) or lemmatization, part-of-speech tagging, dependency parsing. Commonly used corpora already include at least some linguistic annotation in their official distributions.

The official BNC distribution, in the XML-TEI format, is tokenized, lemmatized and part-of-speech tagged with the CLAWS tagger;[18] further annotation concerning structural properties of the texts is also available. If dependency-based DSMs have to be extracted from the BNC, syntactic annotation needs to be added to the original

[17] Kiela & Clark (2014) focus on dependency-based contexts employing the syntactic N-grams from the Google Books (Goldberg & Orwant, 2013).

[18] http://ucrel.lancs.ac.uk/claws/

distribution: for example, Padó & Lapata (2007) employ MINIPAR 0.5,[19] Kiela & Clark (2014) use the parser by S. Clark & Curran (2007), and Baroni & Lenci (2010) annotate BNC with the Tree-Tagger (Schmid, 1995) and MALT parser (Nivre, 2003).

The ukWaC and WaCkypedia_EN corpora are distributed with part-of-speech tagging and lemmatization performed with the Tree-Tagger (Schmid, 1995), and dependency-parsing annotation performed with the MALT parser (Nivre, 2003). This off-the-shelf annotation is commonly used in reference studies (Baroni & Lenci, 2010; Baroni, Dinu, & Kruszewski, 2014). The 2012 Wikipedia snapshot used by Polajnar & Clark (2014) has been annotated with the morphological analyser by Minnen et al. (2001). Kiela & Clark (2014) annotate ukWaC with the parser by S. Clark & Curran (2007).

The original distributions of the Google Ngram corpora are based on tokenized text; since the source text from which they have been extracted is not available, it is not possible to further annotate them. Goldberg & Orwant (2013), however, made available a morphologically and syntactically annotated version of the Google Books Ngram corpus using state-of-the-art tools: a Conditional Random Field (CRF) based tagger (Lafferty et al., 2001; Y. Lin et al., 2012) and a beam-search shift-reduce dependency parser (Zhang & Clark, 2008).

2.3.2 Extraction: from a corpus to a matrix

In the previous section, we described the preliminary steps for the construction of a DSM: selection of the source corpora and pre-processing. In this section, we discuss the design choices related to the extraction of a **co-occurrence matrix** from a corpus (or selection of corpora). This section is structured according to the the theoretical coordinates established by the formal definition discussed in section 2.2, and builds on the following preliminary assumptions:

- We assume the input to the context selection function to be a corpus of precedence graphs encoding surface co-occurrence (cf. figure 2.3) or a corpus of dependency graphs encoding syntactic co-occurrence (cf. figure 2.4).
- We assume the graph nodes to correspond to inflected words, with further annotation specifying lemma, part of speech information, and global frequencies in the whole corpus; this annotation is exploited at the context selection and basis mapping level, and it is accessed by the functions $lemma(node)$, $pos(node)$, $frequency(node)$, $frequency_{lemma}(node)$, $frequency_{pos-lemma}(node)$.
- We assume closed-class words to be part of the sentence graphs, while punctuation is excluded from them.

In what follows, we will characterize the extraction functions outlined in the previous section (context selection, path-value, and basis mapping function) in terms of their theoretical properties and concrete implementations in the DSM literature. Before that, however, another crucial design step which pertains to co-occurrence extraction but is preliminary to the application of the extraction function needs to be characterized: the selection of the set of target terms (T), often referred to as the **vocabulary** of a DSM. At a general level, there are two of possible strategies for the selection of target items: either let evaluation guide target selection and include in T only the items from the evaluated

[19] https://gate.ac.uk/releases/gate-7.0-build4195-ALL/doc/tao/splitch17.html

datasets (Kiela & Clark, 2014), or employ a predefined list of target items, containing all the items from the evaluated datasets and a representative lexical sample of the target language (Baroni & Lenci, 2010; Baroni, Dinu, & Kruszewski, 2014; Levy et al., 2015).[20] Although somewhat disregarded in the DSM literature (DSM evaluation reports often do not explicitly state whether only items from the evaluated datasets were used, or a larger set of words), the selected vocabulary (T) has a notable impact on the output of feature weighting (section 2.3.3.1) and dimensionality reduction (section 2.3.3.2), because they rely on global co-occurrence facts stored in the co-occurrence matrix. As a consequence, restricting the vocabulary to the experimental items would affect feature marginal frequencies or prevent reduction techniques like SVD to capture latent similarities across dimensions; on the other hand, a very large vocabulary increases the complexity of approaches to semantic similarity which require the computation of a distance matrix involving all items in the vocabulary (section 2.3.4).

2.3.2.1 Context selection function

The **context selection function** $cont(t)$ is responsible for the identification of the distributionally informative paths among those anchored at a token for a target t ($\Pi_t, t \in T$). The context selection function $cont(t) = \{\pi \in \Pi_t | \ criteria(\pi)\}$ defines a subset of the paths anchored at each target in the vocabulary, according to quantitative or qualitative **criteria** applied either to the paths or to the nodes at which the paths end.

Let us start from context selection criteria operating on **path properties**. In this domain, path **length** is an unavoidable design choice: for example, the function $cont(t) = \{\pi \in \Pi_t \mid \|\pi\| = 1\}$ selects only paths linking the target to immediately adjacent words in the sentence graph. In bag-of-words DSMs, the context selection function regulates the **size of the context window**; in dependency-based models, it regulates the presence of **mediated dependency relations** holding between targets and context features: for example, paths with $\|\pi\| = 2$ connect the subjects with the objects of their head verbs, while paths with with $\|\pi\| = 3$ connect adjective modifiers of subject nouns with the objects of their head verbs.[21]

Besides path length, context selection criteria can also be guided by qualitative constraints on the edges composing the paths. Within surface-based bag-of-words models, employing qualitative criteria determines the **asymmetry** of the context window:

- $cont(t) = \{\pi \in \Pi_t \mid label(\pi) = \text{LEFT}\}$ selects paths pointing at the word occurring immediately before the target. In figure 2.3, this implementation of the context selection function would select, for the node *barks*, the path: LEFT + *dog*;

[20] Distributional Memory (Baroni & Lenci, 2010) is built from a target set of approximately 30k pos-disambiguated lemmas. The set of words from the evaluated dataset is integrated with a further sample selected on the bases of frequency in the concatenated BNC, ukWaC and WaCkypedia_EN: the top 20k most frequent nouns, and the top 5k most frequent verbs and adjectives. Baroni, Dinu, & Kruszewski (2014) collect distributional vectors for the 300k most frequent words in the same corpora. Levy et al. (2015) build DSMs from a 2013 dump of Wikipedia, ignoring words occurring less than 100 times, resulting in a vocabulary of 189,533 terms.

[21] Padó & Lapata (2007) propose an alternative formulation of the context selection function, which accounts for the window size criterion (k) in surface-based DSMs relying on the positional indexes (*position*) of the start and end node of the path: $cont(t) = \{\pi \in \Pi_t \ | position(start(\pi)) - position(end(\pi)) \leq k\}$. Here, we propose a graph-based implementation of the context selection function which is based on path length and applies to surface- and syntax-based models alike.

- $cont(t) = \{\pi \in \Pi_t \mid \|\pi\| \leq 2 \wedge label(\pi) \in \{\text{LEFT}\}^*\}$ selects the first two words occurring to the left of the target. In figure 2.3, this implementation of the context selection function would select, for the node *barks*, the paths LEFT + *dog*, LEFT LEFT + *cute*;

- $cont(t) = \{\pi \in \Pi_t \mid label(\pi) \in \{\text{LEFT}\}^*\}$ selects the left context of the target, up to the beginning of the sentence. In figure 2.3, this implementation of the context selection function would select, for the node *barks*, the paths: LEFT + *dog*, LEFT LEFT + *cute*, and LEFT LEFT LEFT + *a*;

Within syntax-based DSMs, qualitative criteria on path labels correspond to **linguistic constraints** on the nature of the encoded dimensions. For example:

- $cont(t) = \{\pi \in \Pi_t \mid label(\pi) \in \{\text{AMOD}, \text{SUBJ}\}\}$ selects paths consisting of either an adjectival modifier or a subject. In figure 2.4, this implementation of the context selection function would select two paths: AMOD + *cute*, anchored at *dog*, and SUBJ + *dog*, anchored at *barks*;

- $cont(t) = \{\pi \in \Pi_t \mid label(\pi) \in \{\text{AMOD}, \text{SUBJ}\}^*\}$ selects paths consisting of an adjectival modifier, a subject, or their combinations. In figure 2.4, this implementation of the context selection function would select three paths: AMOD + *cute*, anchored at *dog*, SUBJ + *dog*, anchored at *barks*, and SUBJ AMOD + *cute*, anchored at *barks*;

Besides the properties of the paths, context selection can also operate also on properties of the **nodes** at which the paths end, e.g., their part of speech or frequency. For example:

- $cont(t) = \{\pi \in \Pi_t \mid pos(end(\pi)) \in \{\text{NOUN}, \text{VERB}, \text{ADJECTIVE}\}\}$ selects paths ending at open-class words;[22]

- $cont(t) = \{\pi \in \Pi_t \mid pos(end(\pi)) \in \{\text{NOUN}, \text{VERB}, \text{ADJECTIVE}\} \wedge freq(end(\pi)) > f\}$ selects paths ending at open-class words whose frequency is above a threshold f;

- $cont(t) = \{\pi \in \Pi_t \mid lemma(end(\pi)) \notin S\}$ selects paths ending at words whose lemma does not belong to a list of stop-words (S).

Parameter overview In this section, we review the DSM parameters connected to the manipulation of the context-selection function.

As discussed above, **path length** is a crucial context-selection criterion. Within surface-based DSMs, it determines the size of the context window, a parameter which has been widely explored in the literature on DSM evaluation. In what follows, the terms window size and path length will be used interchangeably. Sahlgren (2006) experiments with π ranging from 1 to 20 at incremental steps of one, while Bullinaria & Levy (2007,

[22] Note that, in this formulation, closed-class words are not selected as context features but they are still taken into account for the computation of the context window. Excluding closed-class words also from the computation of the context window is, of course, a possibility for the construction of bag-of-words DSMs: within the framework described in this section, this design choice is implemented at the pre-processing step (removing the nodes from the sentence graph).

2012) extend the context window up to 100 words.[23] Such a large evaluation scope for π is computationally expensive, and performances are shown to dramatically drop at very large window sizes ($\pi > 10$), and consistently across evaluation tasks. For this reason, DSM evaluation studies tend to either focus on a smaller range, or to set window size to a fixed value to ease comparisons across different model settings and tasks: Kiela & Clark (2014) adopt $\pi = 1, 2, 4, 8$ and full sentence; Baroni, Dinu, & Kruszewski (2014) experiment with $\pi = 2, 5$; Padó & Lapata (2007) and Baroni & Lenci (2010) adopt a window size of 5, while Polajnar & Clark (2014) sets the context window to the full sentence. At a (very) general level, the take-home message of those studies is that smaller window sizes are sufficient for detecting synonymy relations, while larger window sizes bring semantic relatedness into the picture (i.e., words that albeit not interchangeable, are yet distributionally similar because they occur in similar contexts). Within syntax-based DSMs, seminal work (G. Grefenstette, 1994; D. Lin, 1998; D. Lin & Pantel, 2001, and, more recently, Kiela & Clark, 2014) focuses on paths of length one ($\|\pi\| = 1$). Padó & Lapata (2007) experiment with models built from dependency paths up to 4 edges long (finding $\|\pi\| \leq 3$ to be an optimal configuration for many of the involved tasks). Baroni and Lenci (2010) and Rothenhäusler & Schütze (2009) do not manipulate path length as an evaluation parameter, but a closer look at the selected paths reveals that their studies involved paths of length up to three edges.

We now turn to the context-selection criteria operating on **path labels**. Within surface-based DSMs, it regulates the asymmetry of the context-window: evaluation work comparing DSMs built from left vs. right vs. left and right context (Sahlgren, 2006; Bullinaria & Levy, 2007, 2012) showed a clear superiority of symmetric windows over the asymmetric ones. Within syntax-based DSMs, the selection of the distributionally informative contexts is guided by qualitative (linguistic) criteria. While in earlier studies (G. Grefenstette, 1994; D. Lin, 1998) all relations were employed for the construction of the semantic space, later work focuses on a subset of core syntactic relations (e.g., such as subject, object, noun modifiers, prepositional complements, conjuncts) identified either via manual selection (Rothenhäusler & Schütze, 2009; Baroni & Lenci, 2010), or by adopting quantitative criteria on the frequency of the dependency relation in the parsed corpus (Padó & Lapata, 2007). Of the mentioned evaluation studies, only Rothenhäusler & Schütze (2009) further manipulate the context-selection function, by comparing model performance when different dependency relations are involved in the construction of the semantic space (e.g., subjects vs. subjects and objects vs. subjects, objects, and conjuncts, etc.): their results, albeit limited to only one task (noun categorization), clearly show that a very limited set of relations (in their case, object, adjectival modifiers, and conjuncts) already produces the best DSM performances.

Let us now discuss the parameter space of the context-selection functions operating on **end node** properties. The simplest context-selection strategy is to employ the same vocabulary used for the target to define of the set of potential contexts (Baroni & Lenci, 2010; Baroni, Dinu, & Kruszewski, 2014). As an alternative, DSM evaluation targets the application of a frequency threshold (e.g., selecting only words above or below a certain frequency threshold), or the selection of candidate context words based on their rank in

[23] When such large context windows are employed, co-occurrence operates across sentence boundaries. Even though this implementation of the extraction of surface co-occurrences cannot straightforwardly be implemented in our formal account, in which every sentence is represented by either a precedence or a dependency graph, it can easily be accommodated considering the entire corpus as a precedence graph.

the corpus frequency list (e.g., considering only the most frequent n words as potential contexts). Frequency thresholds or ranks are used to filter out words that are too infrequent to significantly contribute to the semantic representations of the targets, or, to the opposite, too frequent and potentially "distributionally promiscuous" (Sahlgren, 2006, p. 105) to be discriminative. Sahlgren (2006) experiments with both minumum (up to 10 occurrences) and maximum (from 5k to 100k occurrences) frequency thresholds: the results, albeit limited to one semantic similarity task and to relatively small corpora (BNC and TASA), show no effect for the minimum frequency, and strong interactions with corpus size for the maximum frequency thresholds. Overall, the frequency rank approach is the most employed in the DSM literature. Bullinaria & Levy (2007, 2012) experiment with up to the 100k most frequent words in the respective corpora (BNC or UkWaC), Kiela & Clark (2014) with up to the 500k most frequent words in the BNC (even if the evaluated corpora are larger). The general trend identified in such studies is that not more than 50k dimensions are usually necessary to ensure good performances, making 50k the reference value for studies which do not evaluate frequency-based context selection (Polajnar & Clark, 2014); for smaller corpora, even fewer dimensions (e.g., 10k) turn out to be sufficient, and more may even have detrimental effects on DSM performance. Under the assumption that words with very high-frequencies just introduce noise and increase the computational cost, with no clear advantage for the resulting semantic representations (cfr. Sahlgren's maximum frequency threshold discussed above), Bullinaria & Levy (2012) test the effect of discarding the 201 most frequent words or using a list of stop-words[24] (which mostly contains function words), and find out that small improvements can be achieved with this context selection technique; they also observe, however, that vector optimization (e.g., context weighting techniques discussed in section 2.3.3) already reduces the impact of the high-frequency words on the semantic representations. Comparable results can also be achieved by combining a threshold on frequency rank and a filter on the part-of-speech of the context words, e.g., by allowing only open-class words as context dimensions.

When it comes to syntax-based DSMs, it is possible to identify two main strategies for the application of frequency thresholds in context selection. The first is to apply the filter to the frequency of the context words (Baroni & Lenci, 2010), before basis mapping creates the dimensions of the semantic space (see section 2.3.2.3). As an alternative frequency thresholds can be applied to the basis elements, after the context dimensions have been created by the basis mapping function (Padó & Lapata, 2007).[25] Note that the application of a frequency filter after basis mapping and frequency aggregation (e.g., based on the frequency of SUBJ+*bark* in the co-occurrence matrix vs.

[24] In addition, Bullinaria & Levy (2012) compare the performances of DSMs built with a context-word stop-list with those of DSs built from a *stopped corpus*, i.e., a corpus from which the words in the stop lists have been removed before co-occurrence extraction, and therefore do not enter into the computation of the size of the context window. Their comparisons do not show any significant advantage for the use of a stopped corpus, besides the speed up in the computations related to a reduced corpus size. DSMs performances, it is observed, only display a shift towards smaller windows. Please note that the use of a stopped corpus, discussed here for ease of comparison with the discussion of stop-lists, is a design choice which pertains to pre-processing and not to context selection.

[25] Padó & Lapata (2007) evaluate DSMs built from the most frequent 500, 1k and 2k basis elements from the BNC, finding optimal performances with the highest threshold. Given that word-based mapping is applied (see section 2.3.2.3 for more details), that the most frequent syntactic relations are selected and that all words are used as targets, we can assume the frequency ranking of basis elements to be reasonably similar to the word frequency ranking among open-class words in the full BNC.

frequency of *bark* in the corpus) belongs, theoretically, to the dimensionality reduction strategies which will be discussed in section 2.3.3.2. In their experiments on the parsed UkWaC, Rothenhäusler & Schütze (2009) do not employ any frequency filter on the basis mapping output.

2.3.2.2 Path value function

The path value function determines the relative contribution of different paths to the quantification of co-occurrence, and it operates on quantitative (length) or qualitative (edge labels) properties of the set of paths identified by the context selection function $cont(t)$.

The default option is to assume that all paths have the same weight, independently of their properties (constant weighting), i.e., $\nu(\pi) = 1$: all words in the context window (e.g., 2 words to the left and to the right of the context), and all dependency paths licensed by the context-selection function have the same weight in the distributional profile for a target word t. As an alternative, the path value function can be employed to assign a stronger weight to basis terms that occur closer to the target. The path value function regulates the parameter which, in the DSM terminology, is referred to as the **shape of the context window**. Figure 2.5 displays the distribution of path values according the most common window shapes:[26]

- the window is rectangular if no weighting is applied (figure 2.5, upper panel):

$$\nu_{\text{rec}}(\pi) = 1$$

- the window is offset rectangular if the closest words to the target (i.e., words whose distance to the target is below a threshold z) are excluded from the context window, and all the other words have the same weight (figure 2.5, second panel from top):

$$\nu_{\text{off}}(\pi) = \begin{cases} 1, & \text{if } \|\pi\| > z \\ 0, & \text{otherwise} \end{cases}$$

- the window is triangular if path weight reduces with distance. Example implementations are:
  - HAL's path-value function (Lund & Burgess, 1996) (figure 2.5, third panel from top[27]):

[26] Example sentence "The rain in Spain stays mainly in the plain", surface co-occurrence, symmetric window of size k=4.

[27] In the original HAL implementation exemplified in Lund & Burgess (1996, p. 204, Table 1), path values range from 1 to the size of the context window, with maximum values assigned to closest words: for example, with $k = 4$, context words immediately adjacent ($\|\pi\| = 1$) are assigned a co-occurrence value of 4. For better comparability with the other weighting schemes discussed in this section, and in accordance with the description of HAL's weighting scheme provided by the authors ("Words within this window are recorded as co-occurring with a strength inversely proportional to the number of other words separating them within the window.", p. 204), in figure 2.5 (third panel from top) we report path weights scaled over the size of the context window.

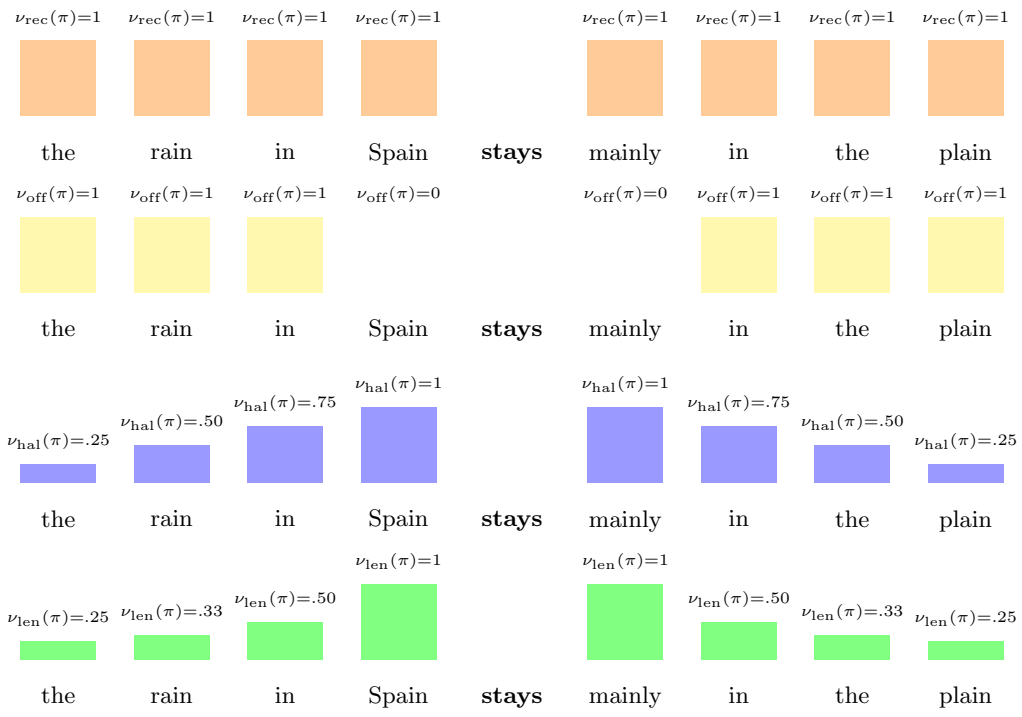

Figure 2.5: Path value function and window shape: rectangular vs. triangular

$$\nu_{\text{hal}}(\pi) = 1 + (k - \|\pi\|)$$

– Inverse path length function (figure 2.5, lower panel):

$$\nu_{len} = \frac{1}{\|\pi\|}$$

A further option, which applies to both surface and dependency-based DSMs is to distribute path weights according to qualitative criteria: for example, in a surface-based DSM, by assigning stronger weights to the left or to the right context; in a dependency-based DSM, by giving more weight to paths containing arguments with respect to those encoding adjuncts.

Parameter overview Within window-based bag-of-words models, the shape of the context window has been widely investigated in earlier work (Lund & Burgess, 1996; Sahlgren, 2006; Bullinaria & Levy, 2007) which led the DSM community to the consensus that a rectangular window ($\nu(\pi) = 1$) is the most reasonable choice in standard semantic similarity tasks.

Within dependency-based models, very little evaluation work on the manipulation of the path-value function has been conducted. The only exception is the evaluation conducted by Padó & Lapata (2007) who experiment with both a quantitative and a qualitative implementation of the path-value function. They compare an unweighted path-value function ($\nu(\pi) = 1$) with two weighted versions of it. The first is a triangular context window based on path length ($\nu_{\text{len}(\pi)} = 1/\|\pi\|$), penalizing paths which encode mediated dependency relations (e.g., from the adjective modifier to a verb, through a modified noun noun). The second is linguistically informed, and relies on a saliency

hierarchy between grammatical relations (subjects are more salient than objects, which are in turn more salient than prepositional phrases and genitives); more specifically, Padó & Lapata (2007) employ the obliqueness hierarchy by Keenan & Comrie (1977) to define a weighting scheme which assigns stronger weights to more salient relations, independently of path length:

$$\nu_{\text{gram-rel}}(\pi) = \begin{cases} 5, & \text{if } \textit{subj occurs in label}(\pi) \\ 4, & \text{if } \textit{obj occurs in label}(\pi) \\ 3, & \text{if } \textit{obl occurs in label}(\pi) \\ 2, & \text{if } \textit{gen occurs in label}(\pi) \\ 1, & \text{otherwise} \end{cases}$$

Evaluation conducted on a benchmark dataset identified in the length-based path-value function the most sensible option for weighting dependency paths, with the linguistically-informed version performing even worse than its unweighted counterpart.

2.3.2.3 Target and basis mapping functions

The **target mapping function**, $\mu_t : T \rightarrow T$ operates on available **node annotation** (e.g., stem, lemma, and/or part of speech) and maps a target token to different versions of it: for example, *dogs* can be mapped to *dogs* (no target mapping on target), *dog* (lemma), or *dog_noun* (part-of-speech disambiguated lemma). DSMs built from inflected words retain semantically-relevant morphological information such as the opposition between singular and plural within the nominal domain, and aspectual information within the verbal domain; moreover, inflectional morphology in some cases does perform an implicit part-of-speech disambiguation (e.g., *booking* and *booked* are verbs). DSMs built from lemmatized data are denser than their inflected counterparts, because all the inflected variants of a target words are conflated. A third commonly used option is to employ part-of-speech disambiguated lemmas, building DSMs which benefit from word-class disambiguation (e.g., *book_noun* vs. *book_verb*), but are sparser than their lemmatized counterparts.

Let us now turn to the **basis mapping** function, which belongs to the original formalization proposed by In Padó & Lapata (2007). The basis mapping function $\mu : \Pi \rightarrow B$ maps the distributionally representative paths for a target t (output of the context selection function) to basis terms (the dimensions of the semantic space) by collapsing paths that are considered equivalent. Basis mapping operates on path labels, node annotations, or both. In turn, basis mapping involving **path labels** can be implemented via a word-based mapping strategy or a structured mapping strategy.

When **word-based mapping** is applied, all paths ending at the same word are considered equivalent, and mapped to the word at which they end: $\mu(\pi) = \textit{end}(\pi)$. This approach to basis mapping characterizes both surface-based DSMs and dependency-based DSMs in which dependency information is used as a co-occurrence filter, i.e., to discard words that co-occur with the target in a sentence, but stand in no syntactic relation with it (Padó & Lapata, 2007). For example, given the context selection function $\textit{cont}(t) = \{\pi \in \Pi_t \mid \|\pi\| = 1\}$, $\mu(\pi) = \textit{end}(\pi)$ would map the target *dog* to the nodes *cute* and *barks* in the surface-based graph in figure 2.3, and to the nodes *cute*, *barks*, and *a* in the syntax-based graph in figure 2.4.

When **structured mapping** is applied, paths are mapped to a concatenation of their label with the word at which they end: $\mu(\pi) = (label(\pi), end(\pi))$. In surface-based models, structured mapping leads to the collection of separate co-occurrence counts for the left and the right portion of the context window (Lund & Burgess, 1996). This choice results in context dimensions of the type LEFT + *cute*, RIGHT + *barks* for the target *dog* in figure 2.3, with $cont(t) = \{\pi \in \Pi_t \mid \|\pi\| = 1\}$. With $\|\pi\| > 1$, structured mapping distinguishes between different positions within the context windows: for example, the target *cute* in figure 2.3 is assigned the features RIGHT+dog and RIGHT RIGHT+barks. To achieve a less sparse representation, it is possible to map paths to a concatenation of their end label with the word they end at: $\mu(\pi) = (label(\pi)_{\|\pi\|}, end(\pi))$. In this case, with , with $cont(t) = \{\pi \in \Pi_t \mid \|\pi\| \leq 2\}$, the target *cute* in figure 2.3 would be mapped to the dimensions RIGHT+dog and RIGHT+dog and RIGHT+barks. In dependency-based models based on structured mapping (G. Grefenstette, 1994; D. Lin, 1998; D. Lin & Pantel, 2001), the context dimensions correspond to features encoding both words and path labels (e.g., AMOD + *cute*, SUBJ-1 + *barks* for the target *dog* from the graph in figure 2.4).

The basis mapping function regulates the trade-off between the **expressivity** of the context dimensions and the **sparsity** of the resulting space. In a window-based DSM the relative position of a context word with respect to a target can be considered an approximation of the syntactic relation holding between the two words; in a syntax-based DSM, the encoding of the syntactic relation in which a specific context word (*dog*) stands with respect to the target (*bark*) allows to tackle finer-grained linguistic tasks such as the modeling of selectional preferences. On the sparsity side, however, structured mapping produces co-occurrence matrices which are sparser than their word-based counterparts (target-context pairs of the type $<$ *bark*, LEFT+*dog* $>$ or $<$ *bark*, SUBJ+*dog* $>$ compared to $<$ *bark*, *dog* $>$). DSMs based on structured mapping are also highly dimensional, and thus computationally more costly: a window-based DSM based on structured mapping (right vs. left of the target word) has twice as many context-dimensions as the corresponding word-based one; a dependency-structured DSM can virtually have a number of dimensions equal to the number of distinct paths (which range from a dozen to hundreds, if the paths are lexicalised) times the number of distinct words at the end nodes (typically ranging from a few to hundreds thousands). It is a matter of evaluation to establish whether the increased computational cost is justified by increased performances.

Basis mapping operating on **node annotation** follows the same criteria discussed for the target mapping function. It can be parametrized to map the path to different instances of the word corresponding to their end node $end(\pi)$: for example, depending on the desired degree of linguistic granularity, the target *dog* is assigned the basis elements RIGHT + *barks* vs. RIGHT + *bark* vs RIGHT + *bark_noun*.

Parameter overview In what follows, we provide an overview of basis mapping approaches that exist in the DSM evaluation literature. Let us start with basis mapping which operates on **path labels** (word-based mapping vs. structured mapping).

Within surface-based DSMs, word-based mapping is the default basis-mapping strategy. There are not many examples of bag-of-words DSMs based on structured mapping, the most notable being HAL (Lund & Burgess, 1996). Bullinaria & Levy (2007, 2012) compare word-based mapping (i.e., total counts for a context word achieved by adding

left and right counts) to structured mapping (i.e., double length vectors with separate counts for left and right context) in a selection of semantic tasks, and do not find a consistent advantage of one approach over the other; the lack of strong empirical support for the use of structured mapping led to the conclusion that word-based mapping, much more convenient from the point of view of the extraction complexity, can be considered a reasonable default option for bag-of-words DSMs. To the best of our knowledge, no further evaluation study produced evidence against this assumption.

Within dependency-based DSMs, word-based mapping produces a filtered (and sparser) version of the corresponding window-based space (Padó & Lapata, 2007), while structured mapping produces a typed version of the dependency-filtered space (G. Grefenstette, 1994; D. Lin, 1998; Rothenhäusler & Schütze, 2009; Baroni & Lenci, 2010). The properties of the two classes of syntax-based DSMs discussed here (filtered vs. typed) have been studied in two types of evaluation settings.

One type of evaluation compares dependency-filtered models (Padó & Lapata, 2007) or dependency-typed models (Rothenhäusler & Schütze, 2009) to window-based models (word-based mapping); in these studies, the performance of syntactically-informed spaces is found to be comparable to or better than the one of bag-of-words models, with variations related to different semantic similarity tasks.

Another type of evaluation targets the degree of path lexicalization in dependency-typed models: Baroni & Lenci (2010) evaluate dependency-typed models characterized by different degrees of path lexicalization, and they extend the comparison to window-based and other state-of-the-art DSMs. A dependency-typed DSM with a very low degree of lexicalization, *DepDM*, is built with path labels corresponding to syntactic relations (e.g., in the sentence "the soldier used a gun" the feature VERB+gun is assigned to the target *soldier*) or prepositions (e.g., in the sentence "the soldier talked with a sergeant" the feature WITH+sergeant is assigned to the target *talk*). A model with a very high degree of lexicalisation, *LexDM*, is built from the same co-occurrence data. *LexDM* is based on complex paths, further articulated into a *pattern* and a *suffix*. The pattern encodes the syntactic path connecting the start (target word) and the end node, the same relations of *DepDM* are encoded, with the crucial difference than high frequency verbs are lexicalised (e.g., USE+gun replaces VERB+gun) and multiword patterns are included (e.g., "such as", "a number of", etc.). The *suffix* encodes the presence of surface features such as determiners, auxiliaries, adjectives or adverbs linked to either the start and end node of the path. For example, in the sentence "the soldier used a gun" the *LexDM* path linking *soldier* to *gun* is USE+*the-n*+*a-n*: the pattern is USE, and the suffix *the-n*+*a-n* specifies that the target is a definite (*the*) singular noun (*n*) and the end node of the path is an indefinite (*a*) singular noun (*n*). Overall, *LexDM* contains more than 3 mln. paths types while *DepDM* contains 800 path types. The high sparsity of the *LexDM* space (0.00001% nonzero entries, vs. the 0.01% of *DepDM*) negatively affects DSM performance, with *DepDM* outperforming *LexDM* in all tasks at issue. Baroni & Lenci (2010), however, show how the rich information encoded in *LexDM*'s paths can be exploited to build a DSM which achieves state-of-the-art performances in all tasks at issue: *TypeDM*. Given that the difference between *TypeDM* and *LexDM* is not in terms of basis-mapping, but in terms of quantification of co-occurrence (co-occurrence counting vs. type-counting), we do not provide more details here and will be describe type counting in more detail in section 2.3.2.4.

Let us now review approaches to basis mapping which operates on **node anno-**

tation. The common approach in DSM modeling is to employ lemmatized words as both targets and contexts. This choice has a practical reason (most evaluation datasets contain lemmatized words) and is supported by evaluation studies. Bullinaria & Levy (2012) compare DSMs built from raw (normalized), stemmed and lemmatized data; in a selection of semantic similarity tasks, a slight advantage is found for lemmatized and stemmed data over the raw ones (with variations due to the different tasks, and to the interaction with other parameters). In a different selection of tasks, Kiela & Clark (2014) compare DSMs whose context words are characterized by different degrees of granularity: inflected, stemmed, lemmatized, pos-tagged, and CCG-tagged:[28] better performances are reported for stemmed data, with no clear advantages for an increased feature granularity. Similar results are reported by (Lapesa & Evert, 2013a), albeit in a more specific task (modeling of semantic priming); no significant differences in performances were found when comparing DSMs built from lemmatized data in three configurations: untagged, with part-of-speech tags on the targets,[29] and with part-of-speech tags for both targets and contexts.

2.3.2.4 Quantifying co-occurrences

In the previous sections, we described the three functions which regulate the extraction of co-occurrence data (context-selection, path value, and basis mapping), and defined their parameter space both theoretically and in practice, with a survey of the DSM evaluation literature. In this section, we characterize the strategies for the aggregation of co-occurrence on the basis of the information collected by the extraction functions. Before discussing the issues connected to the quantification of co-occurrence and the creation of the DSM matrix, let us take stock and spell out the input and the output of each of the extraction functions.

The input of the context selection function $cont(t)$ is a corpus of precedence or dependency graphs (cfr. 2.3 or 2.4, respectively) and a list of targets T. Its output is a set of <*target node*, PATH, *context node*> triples like the following:

$$\{ <\textit{dog}, \textsc{subj-1}, \textit{barks}>, <\textit{cute}, \textsc{amod-1+subj-1}, \textit{barks}>, \text{etc. } \}$$

The path components of the triples produced by the context selection function (e.g., {SUBJ-1, AMOD-1+SUBJ-1}) constitute the input of the path value function, $\nu : \Pi \rightarrow \mathbb{R}$. Based on the path properties discussed in section 2.3.2.2, the path value function assigns to each path a numerical value corresponding to its weight; this information is then integrated into the triple set defined above, to be employed for frequency aggregation after basis mapping. For example, the path value function $\nu(\pi)_{\text{length}} = 1/\|\pi\|$ will produce the following output:

$$\{ <\textit{dog}, \textsc{subj-1}, \textit{barks}, \mathbf{1}>, <\textit{cute}, \textsc{amod-1+subj-1}, \textit{barks}, \mathbf{0.5}> \}$$

The basis mapping function maps paths into basis elements: $\mu : \Pi \rightarrow B$. In practice, it turns the <*target*, PATH, *context*, weight> quadruples produced into <*target*, *basis element*, weight> triples, thus creating the dimensions of the DSM matrix. In our running example, word-based mapping would produce the following set:

[28] CCG tags correspond to lexical categories, and can be considered a finer-grained version of the commonly employed part-of-speech tags.

[29] Formally, the use of part-of-speech tags for the targets does not pertain to basis mapping but to the definition of the target vocabulary, discussed at the beginning of this section.

$$\{ < dog, bark\ , 1>, < cute, bark, 0.5 > \}$$

From the same example, structured basis mapping would produce the following output:

$$\{ < dog, \text{SUBJ-1+}bark, 1 >, < cute, \text{AMOD-1+SUBJ-1+}bark, 0.5 > \}$$

Co-occurrence quantification operates on the set of triples produced by the basis mapping function. **Frequency aggregation** is the most commonly employed approach for the quantification of co-occurrence: given a target t and a basis element b, the co-occurrence frequency of t and b is calculated as the sum of the values for all paths anchored at t which are mapped onto b.[30] In case of structured basis mapping, **marginal frequencies** for feature terms (their global frequencies in the corpus, employed in the computation of association measures as discussed in section 2.3.3.1) are usually computed after basis mapping, and they correspond to the joint co-occurrence frequency of <PATH, *context node*> pairs. Let us consider the following set of tuples:

$$\{ < dog, \text{SUBJ-1+}bark, 40 >, < boss, \text{SUBJ-1+}bark, 10 >,$$
$$< chiwawa, \text{SUBJ-1+}bark, 20 >, < dog, \text{SUBJ-1+}chase, 20 > \}$$

The marginal frequency of the feature SUBJ-1+*bark* amounts to 70, that of SUBJ-1+*chase* to 20. Baroni & Lenci (2010) calculate marginal frequencies separately for PATHs and *context nodes*. In our example tuples the marginal frequency of SUBJ-1 amounts to 90.

A possible alternative to frequency aggregation is **pattern-based co-occurrence** (Baroni & Lenci, 2010), which assigns to the target t and the basis element b a weight which is based on the *number of different paths* linking t and b in the corpus (instead of their plain co-occurrence frequency). This approach to the quantification of co-occurrence is based on the assumption that "the variety of patterns connecting a concept and a potential property[31] is a good indicator of the presence of a true semantic link (as opposed to simple collocational association)" Baroni & Lenci (2010, p. 229). Type Distributional Memory (Baroni & Lenci, 2010) is the only example of a DSM based on pattern-based co-occurrence.[32] Pattern-based co-occurrence falls out of the scope of the general framework defined by Padó & Lapata (2007), and of the present work. It can, however, be accounted for in terms of the formal coordinates discussed in this section: in what follows, we propose a generalization of Padó and Lapata's theoretical framework which allows the repetition of the basis mapping step, which is employed to map paths into basis elements ($\mu : \Pi \rightarrow B$), and, additionally, to map basis elements into new basis elements ($\mu_b : B \rightarrow B$).

Let us start from the Distributional Memory example in section 2.3.2.3: we discussed how, in the sentence "the soldier used a gun", the highly lexicalised *LexDM* maps the target "soldier" to the basis term USE+*the-n+a-n+gun*. *TypeDM* builds on the co-occurrence information stored in *LexDM*, but exploits the pattern/suffix substructure

[30] In the formal definition proposed by Padó & Lapata (2007), a distinction is established between *local* and *global* co-occurrence frequency of t and b. Global co-occurrence frequency measures the co-occurrence of t and b over the entire corpus; it is calculated by summing the local co-occurrence frequencies of b with all instances of t.

[31] "Concept" and "property" in Baroni et al. (2010) correspond to what is referred to in this thesis as "target" and "basis element".

[32] The only other instance of such approach to the quantification of co-occurrence is *Strudel* (Baroni et al., 2010), which is a predecessor of *TypeDM*.

of the paths (in our example, USE is the pattern, and *the-n+a-n* the suffix) to implement a pattern-based approach to the quantification of co-occurrence. To better illustrate the pattern-based approach to co-occurrence implemented in *TypeDM*, let us introduce a few additional sentences containing the target "soldier", and the respective tuples extracted through context selection:

- "the soldier used the gun": <*soldier*, USE+*the-n*+*the-n*, *gun*>
- "soldiers have feelings": <*soldier*, HAVE+*ns*+*ns*, *feeling*>

After the application of the path value (like in Distributional Memory, we assume all path weights to be equivalent) and basis mapping functions frequency aggregation is applied. As a result, a set of co-occurrence tuples is produced, which corresponds to the *LexDM* representation:

{<*soldier*, USE+*a-n*+*the-n*+*gun*, 20>, <*soldier*, USE+*the-n*+*the-n*+*gun*, 10>, <*soldier*, HAVE+*ns*+*ns*+*feeling*, 50 >}

A pattern-based quantification of co-occurrence can be achieved on the basis of the *LexDM* representation, by applying the following steps:

1. The frequency values are transformed via binarization:

   {<*soldier*, USE+*the-n*+*a-n*+*gun*, 1>, <*soldier*, USE+*the-n*+*the-n*+*gun*, 1>, <*soldier*, HAVE+*ns*+*ns*+*feeling*, 1 >}

2. **Secondary basis mapping** is applied to turn basis elements into new basis elements: $\mu_b : B \rightarrow B$. When turning *LexDM* into *TypeDM*, all paths with the same pattern and the same end word are considered equivalent. In practice, basis mapping reduces the paths to their pattern component (USE and HAVE), discards the suffixes, and leaves the context words unaffected. In our running example, the *TypeDM* implementation of pattern-based co-occurrence produces the following output:

   {<*soldier*, USE+*gun*, 1>, <*soldier*, USE+*gun*, 1>, <*soldier*, HAVE+*feeling*, 1 >}

3. Frequency aggregation applies, yielding the count of distinct surface realizations (i.e., *TypeDM* suffixes) instantiating the relation between targets and basis terms:[33]

   {<*soldier*, USE+*gun*, 2>, <*soldier*, HAVE+*feeling*, 1 >}

As an alternative to the binarization approach to the quantification of pattern-based co-occurrence of *TypeDM*, it is also possible (but not explored in the literature) to first apply basis mapping to get rid of the suffix component of the pattern, and then compute

[33]The *TypeDM* implementation requires the calculation of different marginal frequencies for *target*, PATH and *context node*. Marginal frequencies for *target* and *context node* are calculated from the set of tuples resulting from pattern-based frequency aggregation, that of PATH (e.g., USE) is calculated from the set of tuples prior to secondary basis mapping. It corresponds to the number of distinct surface realizations for a path (i.e., the number of different suffixes in which it occurs). In our example, the marginal frequency of the path USE is 2 (i.e., $||\{$*the-n* + *the-n*, *the-n* + *a-n*$\}||$).

co-occurrence by applying a maximum function (output: <*soldier*, USE+*gun*, 20>) or an average function (output: <*soldier*, USE+*gun*, 15>).

In this section, we have discussed two main approaches to the quantification of co-occurrences, namely frequency aggregation on co-occurrence counts and pattern-based co-occurrence. The output of co-occurrence quantification is a set of triples of the form <target, basis term, frequency>, from which a sparse distributional matrix M of shape $T \times B$ is constructed. In principle, M is a full-fledged DSM, and it can already be employed to quantify word similarity as discussed in section 2.3.4. In practice, however, the semantic representation encoded in M can be further optimized and improved by applying a number of operations on co-occurrence counts and on the whole matrix: these will be defined in section 2.3.3 below.

2.3.3 Manipulation of the co-occurrence matrix

In the previous section, we described the process of extraction of co-occurrences from a corpus, as well as the options for the quantification of co-occurrence. The output of the extraction/quantification process is a sparse co-occurrence matrix, which can in principle already be employed for the computation of word similarities (see section 2.3.4). Such matrix is, however, suboptimal with respect to a number of features.

First, the co-occurrence matrix contains raw co-occurrence counts, which are notoriously likely to produce **high frequency effects**. To better characterize a target word t, we are interested in the most *informative* contexts, i.e., those which allow a better discrimination between t and other target words; from this perspective, a very frequent context word is not necessarily the most discriminative one (think about closed class words as determiners, or very unspecific open class words such as *get*, *have*, or *thing*). Section 2.3.3.1 discusses the mathematical operations that can be performed on the co-occurrence data to alleviate frequency bias and "sharpen" the semantic representations: computation of **association measures** which statistically weight the strength of the association between targets and contexts; mathematical transformations applied to feature vectors (i.e., to the columns of the co-occurrence matrix), or to row vectors (to prevent distances from being dominated by a few dimensions, being them the most frequent or most informative); vector thinning (based on frequency or association strength). The common feature of these techniques is that they do not affect the dimensionality of the matrix, but the way co-occurrence strength distributes over target-context pairs. Please note that, in this perspective, matrix sparseness is not a negative feature: as a matter of fact, most of the mathematical operations described in section 2.3.3.1 increase the sparseness of the matrix, thus gaining discriminatory power (Curran & Moens, 2002).

The high dimensionality of the co-occurrence matrix is addressed by applying **dimensionality reduction** techniques, which we review in section 2.3.3.2. Dimensionality reduction can be implemented as **feature selection**: matrix dimensions (columns) are ranked according to a number of criteria (e.g., frequency, variance, number of non-zero entries) and only the top-ranked dimensions are retained for the computation of similarity. Before or after the application of feature selection, further dimensionality reduction is often applied in the DSM literature by resorting to a number of **low rank matrix factorization techniques**, such as Principal Component Analysis, Singular Value Decomposition, and Non-negative Matrix Factorization.[34] The sparse,

[34] Word embeddings (discussed in detail in section 2.4.2) can also be considered a form of dimensional-

high-dimensional representation of the original co-occurrence matrix (in the order of thousands dimensions) is projected into a dense, low dimensional space (in the order of hundreds). Matrix factorization are designed to ensure that at the desired reduced dimensionality, the reduced matrix a good approximation of the original one (that is to say, it preserves most of its variance). The common feature of matrix factorization techniques for dimensionality reduction is that they uncover latent relations among the context dimensions; the resulting reduced dimensions, however, tend to be more opaque than the unreduced ones, and thus more difficult to interpret.

2.3.3.1 Feature weighting and transformation

In this section, we provide an overview of **feature weighting**, a general term which labels a family of mathematical operations performed on the word vectors to highlight the most informative contexts (basis elements) for each target word t. Once feature weighting has applied (or independently of feature weighting, on the raw co-occurrence vectors), a number of further operations can be performed on the vectors to reduce the skewness of co-occurrences (e.g., logarithmic transformation) or to sharpen the distributional representations by increasing the sparseness of the matrix (vector positivization and/or thinning).

Let us start from some general considerations concerning feature weighting. From a **mathematical** point of view, feature weighting operations quantify the strength of the association between the target t and each basis term; they do so by comparing the frequency values stored in the cells of the co-occurrence matrix with reference values which characterize either the basis term by itself (e.g., does it occur with almost all targets?) or the statistical properties of both target and basis term (e.g., how many times would target and basis have occurred together if their association would have been due to chance?). From a **geometrical** point of view, feature weighting operations perform a non-uniform scaling of the components of the vectors which are pulled in the direction of their most discriminative dimensions. From a **cognitive** point of view, weighting operations pick up the most salient features of each target, thereby improving the discriminative power of the distributional representations encoded in the co-occurrence matrix.

Feature weighting has been extensively employed since early work in Distributional Semantics. As discussed in section 2.1.2, early DSMs (primarily LSA) took inspiration from Information Retrieval applications (Salton et al., 1975; Deerwester et al., 1990); such applications are based on a document-term matrix[35] and employ the term-frequency/inverse document frequency measure (henceforth, **tf-idf**) to weight the representativity of a word w for a target document. In the tf-idf scheme, the frequency of occurrence of w in a document (term frequency) is multiplied with the logarithm of the ratio between the total number of documents in the corpus and the number of distinct documents in which w occurs (inverse document frequency). Inverse document

ity reduction. Levy & Goldberg (2014b) show that the embedding matrix can be considered a low-rank factorization of a PPMI matrix shifted by a global constant (see section 2.3.3.1 for more details).

[35]Information Retrieval applications use words as features of documents, thereby building document-term matrix. Their goal is usually to retrieve documents (corresponding to target words in a term-term matrix) based on the similarity between the list of keywords provided by the user (user query is represented as a vector of word) and the vector of the words contained in each document (the feature columns, corresponding to words in a term-term model, or to documents in a term-document model).

frequency quantifies the dispersion of the word across the target documents: the higher the dispersion, the lower the discriminativity. Albeit devised for document-term models, tf-idf can be straightforwardly computed for a term-term matrix: term frequency corresponds to the co-occurrence frequency stored in the matrix cells; inverse "document" frequency is calculated as the logarithm of the ratio between the total number of target words in the corpus and the number of distinct target words with which the feature occurs; an alternative way of quantifying feature dispersion is the number of non zero entries in the matrix columns (G. Grefenstette, 1994). Tf-idf shrinks the less informative dimensions by penalizing the columns of the corresponding contexts. In geometrical terms, this implies that the impact of the less informative dimensions on the position of the distributional vectors is reduced for every target in the distributional matrix.

A desirable outcome for an appropriate feature weighting scheme, however, would be to identify the most representative dimensions on a per-target-and-context basis, and let the dimensions of the most representative contexts for each target determine the position of the corresponding vector in the distributional space. Collocation statistics provides well-established mathematical tools for the quantification of the association between targets and contexts, thereby playing a key role in DSM when it comes to feature weighting. **Association measures** devised to identify true collocates and multiword expressions (Evert, 2008) have become a standard approach for the identification of the most discriminative features for a target word. They quantify the **attraction between a target and a context** by comparing their *observed co-occurrence frequency* (henceforth, O) to their *expected co-occurrence frequency* (henceforth, E). Expected co-occurrence frequency is an approximation of the number of times target (*node*) and context (*collocate*) would have occurred together if their association had been due to chance (i.e., if we had randomly shuffled the words in the corpus), and it is calculated as:

$$E = k \cdot f_{\text{node}} \cdot \frac{f_{\text{collocate}}}{N},$$

where f_{node} and $f_{\text{collocate}}$ are the global frequencies of the node and collocate in the corpus, commonly referred to as *marginal frequencies* in the collocation literature; N is the sample size (number of tokens in the corpus); k is an adjustment factor equal to the total span size (e.g., 10 for a symmetric 5 words context window).[36]

In what follows, we provide a brief overview of the most popular association measures. For a more exhaustive list and discussion, see (Evert, 2008). The most intuitive way of comparing observed and expected frequencies to quantify the amount of evidence provided by O against the null hypothesis of independence (E) is to take the ratio of O/E. This strategy is implemented in the **Pointwise Mutual Information** (PMI)

[36]When calculating expected frequency for surface-based co-occurrence, a span size adjustment is necessary because for every target node there are k slots in which a collocate can potentially occur. See Evert (2008) for a full-fledged description of the calculation of marginal frequencies from contingency tables for node-collocate pairs, and for a discussion of the caveats connected to the span size adjustment. Note that the graph-based approach to surface co-occurrence proposed in this thesis is in principle equivalent to the one outlined in Evert (2008): if marginal frequencies are calculated from the list of target/basis element pairs (i.e., node/collocate) extracted from the sentence graphs, the marginals for both target and basis elements will be inflated by a factor of k, and so will N, keeping E_{graph} equivalent to E calculated according to the standard co-occurrence model defined by Evert (2008).

measure (Church & Hanks, 1990), which quantifies in bits (i.e., on a logarithmic scale) the degree of shared information between node and collocate:

$$\text{PMI} = log_2 \frac{O}{E}$$

PMI distinguishes both attraction and repulsion between a node and its collocate by assigning positive values to the former and negative values to the latter.

PMI has a tendency to assign high scores to low-frequency word pairs with very small E (for example when both feature and target are infrequent, in particular in big corpora where the sample size N is particularly large). Alternative association measures are available within the PMI family, which address the low-frequency bias issue by giving more weight to O in the calculation of the association score. This is the case of Local Mutual Information (LMI) and exponential MI (MI^k), which are calculated as follows:

$$\text{LMI} = O \cdot log_2 \frac{O}{E} \quad \text{MI}^k = log_2 \frac{O^k}{E}$$

Another strategy to counter the low-frequency bias of PMI is to smooth the frequency distribution to increase the value of E. In practice, this is achieved by raising $f_{\text{collocate}}$ and N to a power $\alpha < 1$ (Levy et al., 2015) before computing E.

$$\text{MI}_{\text{smoothed}} = log_2 \frac{O}{E_\alpha} \text{ with } E_\alpha = k \cdot f_{\text{node}} \cdot \frac{f_{\text{collocate}}^\alpha}{N^\alpha}$$

As an alternative to the O/E ratio at the core of the MI family, **z-score** and **t-score** compare the difference between O and E ($O - E$) to E and O, respectively:

$$\text{z-score} = \frac{O - E}{\sqrt{E}} \quad \text{t-score} = \frac{O - E}{\sqrt{O}}$$

Both z-score and t-score distinguish between positive and negative scores; z-score suffers from low-frequency bias, like PMI.

Simple log-likelihood (simple-ll) operationalizes the log-likelihood ratio G^2 (Dunning, 1993) in terms of O and E, and it inherits its robustness to low expected frequencies:

$$\text{simple-ll} = 2 \cdot (O \cdot log \frac{O}{E} - (O - E))$$

Differently from all the association measures listed so far, simple-ll does not distinguish between positive and negative association.[37]

We now turn to **vector transformation** to characterize a family of operations performed on the matrix to further improve the quality of the semantic representations:

- To reduce the skewness of co-occurrences, a **logarithmic** transformation can be applied to raw frequency values (and, in principle, also to the association scores produced by feature weighting). Logarithmic is the most employed (and cognitively motivated) transformation applied to distributional vectors, but other options are available as well, e.g., square root, binarization or sigmoid transformation (a soft binarization);

[37] As pointed out in Evert (2008), the distinction between positive and negative association can be easily re-established by assigning positive scores to node/collocate pairs with $O > E$, and negative scores to pairs with $O < E$.

- To sharpen the semantic representation encoded in the distributional vectors the sparseness of the co-occurrence matrix can be further increased by applying two types of transformations:
  - **Positivization** applies to the output of feature weighting, and capitalizes from the distinction between positive and negative association by setting the negative weights to zero;[38]
  - **Thinning** applies to the raw co-occurrences or to association measures alike: it identifies a subvector for each target by selecting its top-weighted n dimensions and setting all the others to zero.

Summing up, the feature weighting and transformation operations outlined in this section take as an input a matrix M of shape $|T| \times |B|$ and produce a new matrix M_{scored} which has (at least in theory)[39] the same shape of M, is usually sparser than M if positive association measures are employed or, differently from M, may contain negative values (if association measures which distinguish between positive and negative association have been used, without positivization). If we look at M and M_{scored} as multidimensional spaces, feature weighting does not affect the coordinate system of M (the dimensions of M and M_{scored} are the same) but it affects the position of the target vectors in it, because in M_{scored} target vectors are dominated by the most salient dimensions identified by the association measures.

Parameter overview In this section, we provide a summary of the feature weighting schemes and transformations employed in reference evaluation studies.

As far as early DSMs are concerned, HAL (Burgess & Lund, 1998) employs raw co-occurrences, while LSA (Landauer & Dumais, 1997) adopts an entropy-based normalization weighting scheme which can be considered a transposed variant of tf-idf. Term frequency corresponds in LSA to the (log-transformed) frequency stored in the cells of the term-document matrix, and inverse document frequency corresponds to the inverse entropy of the target vector; vector entropy is yet another way of computing context dispersion, as it quantifies the extent to which a target word is a good predictor of the contexts in which it occurs: the same co-occurrence frequency value will have a higher weight for low-entropy targets than in high entropy targets; once again, the higher the entropy, the lower the discriminativity.[40] Albeit popular in earlier DSM work (G. Grefenstette, 1994; Landauer & Dumais, 1997; Sahlgren, 2006) and still present in more recent evaluation studies (Kiela & Clark, 2014), the tf-idf family[41] fell out of the

[38] For example, positive PMI is calculated as: $PPMI = max(PMI(node, collocate), 0)$

[39] Positive association measures and vector thinning do not, theoretically, affect the overall shape of the co-occurrence matrix because they operate in a per-target fashion. It is possible, however, that some context dimensions will turn out to be negatively associated to each target and therefore discarded by positive association measures; it is even more likely that some of the dimensions will turn out to be not among the top n of any target, and therefore discarded by vector thinning.

[40] Landauer & Dumais (1997) point out that multiplying frequency by inverse entropy "accomplishes much the same thing as conditioning rules such as those described by Rescorla & Wagner (1972), in that it makes the association better represent the informative relation between the entities rather than the mere fact that they occurred together" (p. 208).

[41] Here, we use the label "tf-idf family" to refer to a group of weighting schemes which weight co-occurrence frequency against dispersion measures for the columns (G. Grefenstette, 1994; Sahlgren, 2006) or rows of the matrix (Sahlgren, 2006)

focus of the DSM community, which converged on the use of collocation statistics for the purpose of feature weighting. In this perspective, evaluation studies targeted the identification of the best performing association measure among those put forward in the corpus linguistics literature.

In their evaluation study on bag-of-words models, Bullinaria & Levy (2007) compare conditional probability based on raw co-occurrences (corresponding to L1 normalization, see section 2.3.4.1) to PMI, ratio of probabilities (unlogged PMI) and positive PMI (negative weights are set to zero), finding that the latter ensures the best performances in semantic similarity tasks.

Within dependency-based models, little evaluation work has been conducted which compares different weighting schemes: Rothenhäusler & Schütze (2009) compare g-score (log-likelihood) to t-score and positive t-score (negative weights are set to zero). Positive t-score turned out to be the best performing measure. Larger scale evaluation studies focus on just one measure: Padó & Lapata (2007) employ simple log-likelihood, Baroni & Lenci (2010) use LMI.[42]

A number of recent studies confirmed that positive PMI (henceforth, PPMI) is the best association measure across tasks in the following comparison settings:

- Polajnar & Clark (2014): PPMI vs. t-test (calculated as in the z-score formula given above, comparable performances to PPMI) and raw frequency;
- Kiela & Clark (2014): PPMI vs. raw frequency, MI, t-test (calculated as in the z-score formula given above, comparable performances to PPMI), chi-squared (Curran, 2003), and a number of measures from the tf-idf family;
- Baroni, Dinu, & Kruszewski (2014): PPMI vs. LMI.

PMI is nowadays considered the best performing weighting scheme, and it is an established finding that discarding negative weights (PPMI) improves DSMs performances robustly across tasks. Further evaluation work conducted by Levy et al. (2015) targeted the improvement of PPMI by testing two additional parameters:

- **Smoothing context distributions**: experiments on the manipulation of α in the MI_{smoothed} formula given above showed that setting α to values smaller than 1 improves PPMI performance robustly across tasks;
- **Shifting PPMI values** by a global constant: experiments conducted on the manipulation of a global constant k to be subtracted to the PMI values[43] showed that setting k to values above 1 improves DSM performances in word similarity tasks. Shifting PPMI by a positive value results in a sparser matrix than the corresponding PPMI one, because it increases the number of values which are set to zero. For this reason, it can also be seen as a form of thinning.

[42]Note that Baroni & Lenci (2010) calculate a "three-way" LMI, i.e., by comparing O to the corresponding expected count under independence for the triple <*target*, *dependency path*, *collocate*> (e.g., <*dog*, SUBJ-1, *bark*>). Refer to section 2.3.2.4 for more details on the calculation of marginal frequencies for feature terms. In the experiments presented in this thesis (section 4.3), we follow a more standard approach to collocation analysis, in which collocates are basis elements, and calculate expected counts for the <*target*, *basis element*> pairs (e.g., <*dog*, SUBJ-1+*bark*>).

[43]Shifted PPMI is calculated as: $SPPMI(node, collocate) = max(PMI(node, collocate) - log\ k, 0)$.

PPMI achieves significant improvements by discarding the negatively weighted dimensions which are replaced by additional zero entries. Building on this established property of PPMI (and positive association measures), further evaluation by Polajnar & Clark (2014) tested the hypothesis that, even among the positively weighted dimensions, not all of them are necessarily beneficial for the quality of the semantic representations. Additional improvements can be achieved by **thinning** the vector by retaining only the n highest weighted context words. The experiments carried out by Polajnar & Clark (2014) on the manipulation of n show that a surprisingly small number of dimensions need to be retained, with variations across the different weighting schemes: out of 10k of the original DSM, 20 dimensions per vectors were sufficient with raw frequencies, 140 for t-test (z-score), 240 for PPMI.

2.3.3.2 Dimensionality reduction

The large dimensionality of M_{scored} can be a problem for further applications which take DSM vectors as an input, e.g., computation of vector similarity (which is computationally more expensive at high dimensionalities) or machine learning applications (where separate weights need to be learnt for each dimension of the matrix). Moreover, the dimensionality of M_{scored} is problematic at a qualitative level, as well, as M_{scored} is likely to contain noise and redundancy. Noise is introduced by contexts whose variation cannot be explained in terms of their relation to the target words; redundancy is introduced by clusters of highly correlated dimensions (because the corresponding contexts occur with the same targets).

Dimensionality reduction filters away redundancy and noise by reshaping the multidimensional semantic space, i.e., by transforming its coordinate system. There are two strategies to reduce the dimensionality of a multidimensional space: **feature selection**, which discards one or more columns deemed less relevant based on their mathematical properties (e.g., variance, number of non-zero entries, or overall frequency); **feature extraction**, which identifies clusters of highly correlated dimensions, builds a statistical summary of the co-occurrence information encoded in these dimensions, and replaces them with this summary (a new dimension which does not correspond to any observed context).

Let us start with a toy example in a two-dimensional space with coordinates x and y (figure 2.6), and let us assume we want to reduce the space to a one-dimensional representation.

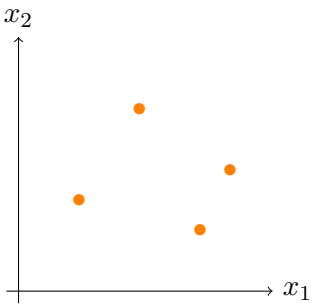

Figure 2.6: Toy space

In our toy space **feature selection** can pick either x_1 or x_2 (figure 2.7); once one of the two dimensions is selected, the points in the space are projected on it (dotted blue line for dimension x_1, dotted red line for dimension x_2): the projection, straightforward in this case, assigns a value which encodes its position on the selected dimension, and the other dimension is not any longer taken into account. In the high-dimensional space defined by the co-occurrence matrix, this amounts to discarding some of the matrix columns.

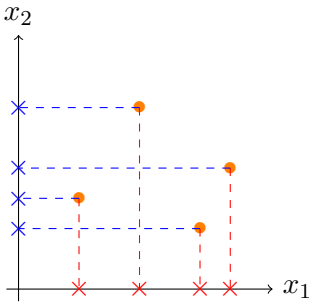

Figure 2.7: Dimensionality reduction by feature selection

Figure 2.8 illustrates the **feature extraction** procedure for our toy space. A new dimension is introduced which is considered an appropriate replacement of x and y (the criteria that the new dimension must match are a property of the dimensionality reduction algorithm, and will be discussed later on in this section). The points are projected on the newly introduced dimension (the red line), x_1 and y_2 are discarded, and each point is now defined in terms of its position on the red line.

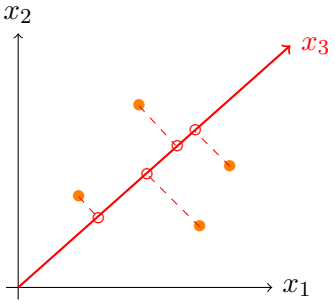

Figure 2.8: Dimensionality reduction by feature extraction

Dimensionality reduction by feature extraction is implemented by employing **low rank matrix factorization** techniques which project a multidimensional space $X_{m \times n}$ into a space of a reduced dimensionality $Y_{m \times d}$ with $d \leqslant n$.[44] Matrix factorization algorithms are designed to ensure that, at the desired dimensionality d, Y is a good approximation of X "in some sense". The criteria for what it means to be a good approximation differ across factorization algorithms; geometrically, for example, a suit-

[44]Technically, d also has to be smaller than the rank of the matrix X (i.e., the number of linearly independent columns or rows in X).

able criterion would be that the projected points are as close as possible to the original points.

Principal Component Analysis (PCA) and **Singular Value Decomposition** (SVD) implement the projection of X into Y with the tools of statistical analysis and linear algebra, respectively. PCA and SVD can be seen as methods for describing the original data *in an alternative way*: they identify a ranked list of directions which do a better job at capturing the variance in X than the original dimensions. The new directions are the dimensions of Y, referred to as **eigenvectors** or **principal components** in PCA and as **singular vectors** in SVD. Principal components and singular vectors are orthogonal. In geometrical terms, the mapping between X and Y is implemented as an orthogonal projection followed by a rotation of X over Y by converting the position of each point (i.e., the row vectors in X) from the coordinate system of X into the "condensed" coordinate system of Y.

Figure 2.9 (Jurafsky & Martin, in press) illustrates the application of PCA to a bi-dimensional space, displayed in panel (a). The first principal component, PCA dimension 1 in panel (b), is the direction which accounts for most of the variance in the original space. The second principal component is the direction orthogonal to PCA1 which accounts for most of the variance left unaccounted. Panel (c) shows the result of the rotation from the original space to the space which has PCA1 and PCA2 as coordinates. Keeping both principal components reconstructs the original space, while selecting only the first one produces an *approximation* of it and performs dimensionality reduction.

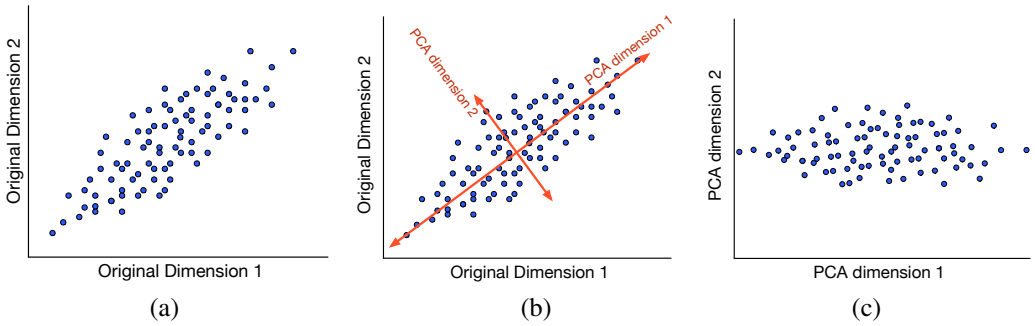

Figure 2.9: Principal Component Analysis

SVD is mathematically equivalent to PCA and it is often employed to perform PCA decomposition more efficiently.[45] A key difference between SVD and PCA is that SVD skips the centering of features required by PCA, and can thus be applied to a sparse matrix.

SVD's low-rank projection is achieved by factorizing the $M_{w \times c}$ into three further matrices, as shown in figure 2.10.

- W: a column-orthonormal matrix of shape $w \times d$, containing the left singular vectors;

[45] SVD is more efficient than PCA because it does not require the computation of the full covariance matrix. A detailed discussion of the mathematics of PCA/SVD will not be provided here, because their comparison is not in the scope of this thesis.

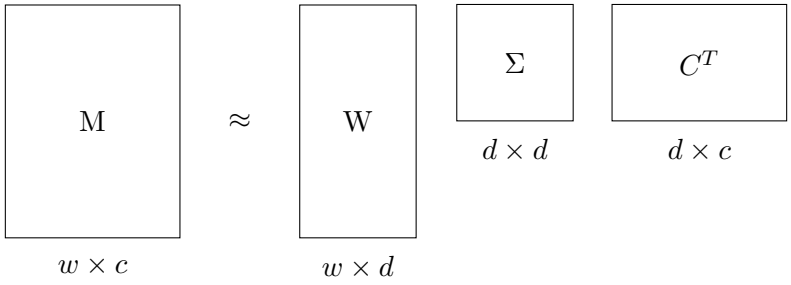

Figure 2.10: SVD low-rank matrix factorization

- Σ: a diagonal matrix of shape $d \times d$ containing the ranked singular values of M (which quantify the variance of the dimensions of W)

- C: a column-orthonormal matrix of shape $c \times d$, containing the right singular vectors.

The multiplication of W, Σ, and C^T produces the best approximation (in the least-square sense) of the original matrix M at the desired dimensionality d. Reduced "latent" vectors for the targets in M are computed by multiplying W with Σ (i.e., by scaling the left singular vectors by the singular values in Σ). Differently from PCA, the first dimensions of $W \times \Sigma$ don not capture the largest amount of variance but instead they constitute the best approximation of M at the desired dimensionality.

SVD (and PCA) have been criticized because they negatively affect the transparency of the semantic dimensions by **destroying their non-negativity** (it is unclear what negativity on a semantic scale stands for) and **turning the sparse representation into a dense one**. Further criticism to SVD targets its treatment of high frequency words, which receive either too much or too little weight, and the excessive weight put on the many zero cells of the sparse matrix (Levy & Goldberg, 2014a; Levy et al., 2015).

Various other factorization techniques are available, which address these shortcomings, the most known example being **non-negative matrix factorization** (NMF). NMF (Lee & Seung, 2000) enforces that the matrices in the factorization contain elements greater or equal to zero, producing reduced vectors which are still sparse and therefore more interpretable. For a discussion of experimental results of NMF in distributional modeling see Dinu & Lapata (2010) (bag-of-words) and Van De Cruys (2010) (dependency-based).

Besides SVD and NMF, further low-rank matrix factorization techniques are available, such as GloVe (Pennington et al., 2014) or the word embeddings discussed in section 2.4.2. Given that the experimental focus of this thesis is kept on the popular and efficient SVD, we do not further elaborate further on these techniques.

Parameter overview In this section, we briefly sketch the DSM evaluation literature with respect to dimensionality reduction strategies.

As far as **feature selection** is concerned, it is necessary to stress the difference between the dimensionality reduction operations discussed in this section and context selection based on global frequency values discussed in section 2.3.2.1. Feature selection

operates on the co-occurrence matrix *after* basis mapping and co-occurrence quantification. It is therefore likely to produce different results than context selection if a restricted target vocabulary is employed, in case of structured mapping and dependency filtered bag-of-words models where only a subset of the available syntactic relations are allowed by the context selection function (Padó & Lapata, 2007). By inspecting the DSM literature, it is not always easy to disentangle whether reported experiments should be classified as context selection or dimensionality reduction. In what follows, we discuss a few cases for which it is clear that feature selection was performed on the co-occurrence matrix, after basis mapping. Within bag-of-words models, Burgess & Lund (1998) apply feature selection based on column variance after the structured basis mapping characterizing HAL (whose co-occurrence matrix distinguishes between right and left contexts); they find that most of the effects on DSM performance are due to the 100/200 most variant dimensions (out of 140k).[46] Kiela & Clark (2014) operate feature selection by sliding a 10k window across the columns of a 50k bag-of-words co-occurrence matrix, trying to identify the most optimal "contiguous subvector" in terms of frequency range of the selected contexts; they find out that, on matrices containing raw co-occurrence frequencies, focusing on low frequency contexts may improve DSM performance in some tasks. Within dependency-filtered bag-of-words models Padó & Lapata (2007) explicitly report on feature selection based on basis terms frequency (best configuration on their development task: 2000 basis elements from the BNC).

When it comes to dimensionality reduction by **feature extraction**, SVD plays the key role since early DSM work. Landauer & Dumais (1997) employ SVD in LSA, pointing out that this method does not only make the space more manageable by reducing the size of the term-document matrix, but that it also improves the quality of the semantic space by inducing latent relations holding among the contexts encoded in the matrix dimensions (in LSA term-document setting, SVD dimensions encode latent document topics). As far as the parameter space is concerned, there are two main (and connected) parameters which have been explored, namely:

- The **number of reduced dimensions** retained from the full SVD space.
  Since LSA (which employed 300 reduced dimensions), the DSM community converged on the tendency to retain only a few hundred dimensions from the reduced matrix. Bullinaria & Levy (2012), however, conducted extensive experiments on the dimensionality of the reduced matrix showing that: a) on term-term models the improvement achieved by applying SVD is less dramatic than reported with LSA; b) in order to improve DSM performance on term-term model, SVD usually requires more than the "standard" 300 dimensions; further improvements can be achieved by discarding the first reduced dimensions, associated to the strongest singular values and thus having the highest variance.
  Baroni, Dinu, & Kruszewski (2014) compare SVD and NMF in set of standard task, and at a range of reduced dimensions ranging from 200 to 500, and found better performances at the higher dimensionalities, with SVD clearly outperforming NMF.

[46] Rohde et al. (2006) point out that "as the magnitude of a set of values is scaled up, the variance of that set increases with the square of the magnitude. Thus, it happens to be the case that the most variant columns tend to correspond to the most common words and selecting the k columns with largest variance is similar in effect to selecting the k columns with largest mean value, or whose corresponding words are most frequent increases with the square of the magnitude."

- The manipulation of the **singular values** in the computation of the reduced vectors WS.
  Caron (2001) found out that raising Σ to a power p smaller than 1 (thereby emphasizing later components, with smaller singular vectors) improves the effect of SVD on LSA. This result has been replicated by Bullinaria & Levy (2012) and Levy et al. (2015) for term-term models.

2.3.4 Projecting meaning in space

In the previous sections, we discussed and motivated the DSM design choices connected to the extraction of co-occurrence information and to the enhancement of the semantic representation via frequency weighting or dimensionality reduction. In this section, we discuss the quantification of the semantic similarity on the basis of the contextual information stored in DSM vectors. It is at this stage that the abstract notion of **meaning similarity** is translated into the geometric notion of **distance** and thus empirically quantified. In section 2.3.4.1 we provide a taxonomy of the distance metrics which are most commonly employed in the DSM literature, and we discuss the symmetric vs. asymmetric contrast which characterizes these measures. Section 2.3.4.2 presents an alternative approach to the computation of word similarity which is still based on vector distances but takes a finer-grained perspective on the network of relations in which target words are placed. It quantifies relatedness in the semantic space based on properties of the **semantic neighborhood** of the target t: in practice, this means that to know how similar t and t_1 are it is not sufficient to compare $\vec{t}$ and $\vec{t_1}$, but we need to calculate their similarities to all targets in the DSM vocabulary. Such approaches, based on neighbor ranks or graphs, are inherently asymmetric, and thus of particular interest when it comes to modeling cognitive processes which are notoriously not symmetric.

2.3.4.1 Distance measures

Broadly speaking, there are two approaches for the quantification of meaning similarity based on the co-occurrence information encoded in a DSM. The first approach is grounded in the geometric interpretation of meaning (Widdows, 2004): DSM vectors are seen as a set of coordinates in a multidimensional space and meaning similarity is quantified as the distance between vectors in this space. The second approach is grounded in information theory: DSM vectors are interpreted as probability distributions and compared in terms of their shared information content.

The geometric approach to the quantification of meaning similarity is implemented by employing distance metrics: a **distance metric** $d(x, y)$ is a function which maps pairs of points (x and y) into real values quantifying how far apart the points are in the multidimensional space in which they live; in a DSM application, the closer the points, the more similar the meanings of the corresponding words.

Given a set of points X, distance metrics satisfy the following properties for all x, y, z in X:

- non-negativity: $d(x, y) \geqslant 0$;
- symmetry: $d(x, y) = d(y, x)$ (Tversky, 1977);
- coincidence: $d(x, y) = 0 \iff x = y$;

- triangle inequality: $d(x, z) \leqslant d(x, y) + d(y, z)$

The **cosine** coefficient, the most commonly employed measure in the quantification of meaning similarity, is not a distance metric. Given two vectors $\vec{u}$ and $\vec{v}$, cosine similarity is calculated as follows:

$$s_{cosine}(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{i=n} u_i \cdot v_i}{\sqrt{\sum_i u_i^2} \cdot \sqrt{\sum_i v_i^2}}$$

The computation of cosine similarity is based on the inner product between $\vec{u}$ and $\vec{v}$: it measures the overlap between the dimensions of $\vec{u}$ and $\vec{v}$, as any zero entry in either $\vec{u}$ or $\vec{v}$ sets the corresponding product to zero. Other similarity measures from the inner product family are the Jaccard and the Dice coefficients, and the Lin's measure.[47] Similarity measures which range from 0 to 1 (cosine is bound to $[0, 1]$ in the positive space) can be turned into dissimilarity measures by taking their complement to 1, i.e., $d = 1 - \{s_{cosine}, s_{Jaccard}, s_{Dice}\}$; note that dissimilarity measures cannot be considered as distance metrics, as they often don't satisfy the triangle inequality. A further possibility which exists only for cosine similarity is to convert it to **angular distance**, which is a full-fledged distance metric: $cos\ \alpha(\vec{u}, \vec{v}) = s_{cosine}(\vec{u}, \vec{v})$

Distance metrics from the **Minkowski family** are based on the (absolute) pairwise differences between corresponding dimensions of $\vec{u}$ and $\vec{v}$. For $p \geq 1$, Minkowski distance is calculated as follows:

$$d_p(\vec{u}, \vec{v}) = \sqrt[p]{\sum_{i=1}^{n} |u_i - v_i|^p}$$

Well-known members of the Minkowski family are the Manhattan or taxi-cab or L1 distance ($p = 1$) and the Euclidean or L2 distance ($p = 2$).[48] Differently from cosine, these metrics are sensitive to vector magnitude: for this reason, it is is crucial that vectors are **normalized** (i.e., converted to unit length) by employing the compatible **norm** (the length of the vector, i.e., the distance of the point from the origin of the axes):

$$\|\vec{u}\|_p = \sqrt[p]{\sum_{i=1}^{n} |u_i|^p}$$

For example, L1 normalization adjusts feature weights such that $\sum_{i=1}^{n} |u_i| = 1$, effectively turning the input into a probability distribution.

Let us now turn to the information theoretic view on similarity, which inspires a number of distance measures (not metrics) based on Shannon's notion of probabilistic

[47] Albeit based on information-theoretic considerations, Lin's measure (D. Lin, 1998) is classified here because it is based on the intersection between $\vec{u}$ and $\vec{v}$, and thus connected to the inner product family: "the similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are". The inner product of $\vec{u}$ and $\vec{v}$, as well as their harmonic mean can also be employed to quantify similarity. Note that Jaccard, Dice and Lin coefficient only make sense for non-negative spaces with transparently paired dimensions and are therefore not suitable for reduced vectors.

[48] Maximum (or Canberra) distance is a special case of the Minkowski family with $p \to \infty$; the distance between u and v is calculated as the highest value among pairwise differences of the corresponding dimensions: $d_{p\to\infty}(\vec{u}, \vec{v}) = max|(u_i - v_i)|$. Inclusion measures (see Lenci and Benotto, 2011 for an overview), which quantify the extent to which a vector is a subset of another vector, can be considered as an asymmetric, non metric member of the Minkowski family based on the $min(\vec{u}, \vec{v})$ function and on the L1 norm of $\vec{u}$.

uncertainty or entropy. The most known member of the **information theoretic family** is the Kullback-Leibler divergence, which is asymmetric: two words a and b are similar if the probability distribution of a is a good approximation of the probability distribution of b. Given two vectors $\vec{u}$ and $\vec{v}$, KL divergence assumes $\vec{u}$ and $\vec{v}$ to be in form of a probability distribution (non-negative and L1 normalized), which we will denote as U and V. It is calculated as follows:

$$d_{KL}(\vec{u}, \vec{v}) = D(U\|V) = \sum_{i=1}^{n} u_i \cdot log_2 \frac{u_i}{v_i}$$

KL divergence is problematic for DSMs because it is equal to ∞ whenever a 0 in the probability distribution $\vec{v}$ corresponds to a non-zero in $\vec{u}$. Jensen-Shannon divergence and α-skew divergence are designed to address this issue, the former with a symmetric result, the latter with an asymmetric result.

Parameter overview In this section, we provide an overview of the parameter space responsible for the computation of distance/similarity in a distributional space. Since early work in distributional semantics, the geometric approach to the computation of similarity has been preferred to the information theoretic one: LSA employs cosine similarity, while HAL resorts to the Minkowski family with $p = \{1, 2\}$ (Manhattan and Euclidean distance).

More recent DSM evaluation work converged on *cosine* as the most robust choice for the computation of similarity; besides being robust across tasks and evaluation settings, cosine is also adopted because it is computationally more efficient.

Within bag-of-words models:

- Bullinaria & Levy (2007) compare cosine to Euclidean and City-block distance, KL divergence and other measures from the information-theoretic family;
- Padó & Lapata (2007) compare cosine to Lin coefficient;
- Kiela & Clark (2014) compare cosine similarity to a number of other measures from the inner product family (correlation, Dice, Jaccard, Tanimoto and Lin coefficient), from the Minkowski family (Manhattan, Euclidean and Maximum distance) and from the information theory family (Jensen-Shannon and α-skew divergence);[49]
- Polajnar & Clark (2014) compare cosine to Jaccard and Lin coefficient;
- Bullinaria & Levy (2012), Levy et al. (2015), and Baroni, Dinu, & Kruszewski (2014) simply adopt cosine as a similarity measure.

Within syntax-based models:

- Padó & Lapata (2007) evaluate cosine, Euclidean and Manhattan distance, Jaccard and Lin coefficient, KL and α-skew divergence; based on results on a development task, they adopt Lin similarity measure (D. Lin, 1998) in their main experiments;

[49] In the evaluation conducted by Kiela and Clark (2014), cosine is the most robust measure together with correlation and Tanimoto coefficient.

- Baroni & Lenci (2010) and Rothenhäusler & Schütze (2009) adopt cosine similarity without further evaluation;

Besides evaluation work conducted on specific tasks, the comparison carried out by Weeds et al. (2004) is of particular interest. Their study targets a comprehensive set of distance/similarity measures from the three families described in the previous section, quantifying the variation in the nearest neighbours determined by the chosen measure in terms of overlap among the top neighbors for 2000 target nouns (dependency filtered DSM, direct object relation, BNC). They find substantial differences between the sets of neighbors defined by different measures. Furthermore, they address the questions of whether certain distance measures are more biased towards high frequency words than others, and whether the frequency of the target noun itself influences the set of generated neighbors. They identify a set of measures which show a tendency to select high frequency words as nearest neighbors regardless of the frequency band of the target, namely cosine similarity, Jensen-Shannon divergence and α-skew divergence; they also identify a set of measures which show a tendency to generate set of neighbors from a comparable frequency band of the target noun, namely Jaccard and Lin coefficient.

2.3.4.2 Relatedness in the semantic space

In distributional semantic modeling, similarity between words is calculated according to Euclidean geometry: the more similar two words are, the closer they are in the semantic space. As discussed in the previous section, one of the axioms of spatial models is symmetry (Tversky, 1977): the distance between point a and point b is equal to the distance between point b and point a. Cognitive processes, however, often violate the symmetry axiom: for example, asymmetric associations are often found in word association norms (Griffiths et al., 2007) and similarity ratings (Lapesa, Schulte im Walde, & Evert, 2014). In this section, we discuss two approaches to the quantification of similarity/relatedness in the semantic space which are based on distance/similarity but can also capture asymmetries thanks to their sensitivity to the topology of the high-dimensional distributional space. The basic intuition behind these approaches is that to characterize the similarity between a word w_1 and a word w_2 the distance between $\vec{w_1}$ and $\vec{w_2}$ is not sufficient, but rather it is necessary to consider the network of similarities of w_1 with respect to *all* the other words in the lexicon (i.e., the vocabulary of the target DSM) or at least with respect to the top n most similar ones (in case the full vocabulary is too large).

The first approach is based on **neighbor rank**: given a word pair w_1 and w_2, the degree of relatedness between w_1 and w_2 can be quantified as the position of w_2 in the ranked list of neighbors of w_1. Crucially, the same *distance* can correspond to a different position in the ranked neighbors of a target word, as shown in the toy example in figure 2.11. The symmetric relation encoded in the angular distance between *knife* and *lancet* (left panel) can be made asymmetrical if the density of the neighborhood is taken into account; the same angular distance corresponds in this case to different rank values, with *knife* being the third neighbor of *lancet* (central panel), but *lancet* being only the fifth neighbor of *knife* (right panel) because of higher number of intervening neighbors.

Neighbor rank assigns higher values to unrelated words, ranging from 0 or 1 (the rank of a target word in its own ranked neighbors) to the size of the DSM vocabulary; thus it obeys to the non-negativity constraint and the coincidence constraint discussed

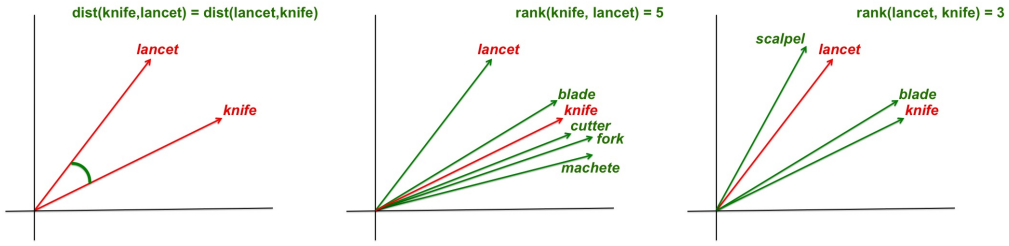

Figure 2.11: Vector distance vs. neighbor rank

in the previous section. Besides symmetry, it also violates triangle inequality. The potential of neighbor rank in cognitive modeling has already been tested by Hare et al. (2009) and Lapesa & Evert (2013a); Zeller et al. (2014) successfully employed it for the quantification of the semantic relatedness between derivationally related words. One of the main contributions of the present thesis is to extend the evaluation of neighbor rank to standard tasks in distributional semantic modeling.

Building on the intuition that distance between two points in a distributional space is not fully satisfactory, Cuba Gyllensten & Sahlgren (2015) propose to employ **neighborhood graphs** as an alternative for the quantification of word similarity or relatedness. Neighborhood graphs allow the identification of the local structure (topology) of the network of relations defined by the nearest neighbors of the target words in the DSM vocabulary. Cuba Gyllensten & Sahlgren (2015) employ *relative neighborhood graphs* (RNG) in a Word Sense Induction task (WSI), which is otherwise usually tackled by inspecting the top k nearest neighbors of a target word (k-NN). k-NN approaches face two main issues: the fact that k needs to be set as a further experimental parameter and the possibility that a ranked list of top k neighbors will conflate the different senses of a polysemous target word. According to Cuba Gyllensten & Sahlgren (2015), a RNG-based approach has the potential of overcoming both issues. In an RNG, two points are considered neighbors only if the region between them is empty:[50] in practice, this means that two words are considered neighbors only if they are the closest neighbors of each other. Their experiments shed light on the distribution of neighborhood reciprocity in different DSM architectures (PMI vs. embeddings) and show that embeddings tend to produce more asymmetric (and syntagmatic) neighborhoods than unreduced PMI spaces. For further details concerning the comparative evaluation of k-NN and RNG in the WSI task, see Cuba Gyllensten & Sahlgren (2015).

2.4 DSM representations based on signal vectors

Section 2.3 discussed the class of bag-of-words DSMs, which are in the focus of the present thesis. Even though bag-of-words models are a commonly adopted methodology for the extraction of distributional representations, an alternative approach exists, namely one that relies on **signal vectors**. This section discusses three types of DSMs based on signal vectors and explains the reasons why such DSMs are outside the scope of this work.

[50] "A point b lies between two points a and c if it is closer to both a and c than they are to each other" (Cuba Gyllensten & Sahlgren, 2015, p. 2454).

Like in the case of bag-of-words models, DSMs based on signal vectors collect co-occurrence information for a target word t by scanning the corpus with a sliding window, which defines representative contexts for t. Yet differently from bag-of-words DSMs, they assign two vectors to each word w:

- A **signal vector**, which is initialized the first time w is encountered, and does not change during training. Static signal vectors contain a random sequence of numbers (usually zeros and ones) which plays the role of a unique identifier for w in the training process, but bears no relation with the semantics of w. Signal vectors are orthogonal or nearly orthogonal to each other and, from a cognitive point of view, interpreted as a static representation of the invariant properties of a word (e.g., its phonetic or orthographic representation).
- A **memory vector**, which characterizes the semantics of w as a target. The memory vector for w is updated every time w is encountered in the corpus, and it corresponds to the output of a training process which takes the signal vectors of the words co-occurring with w as an input. Memory vectors are dense, and their dimensionality is lower than the one of state-of-the-art bag-of-word models: for this reason, the models reviewed in this section can also be seen as methods for dimensionality reduction (cfr. section 2.3.3.2).

Co-occurrence information is not encoded in terms of frequency values stored in the cells of a high-dimensional, sparse matrix (as in bag-of-words models). Instead, it is stored in a dense, low-dimensional vector representation: the **embedding matrix**, which contains the memory vectors. The DSMs described in this section adopt different strategies for the extraction of memory vectors based on the signal-vector representation of context words,[51] which are built either by **accumulating signal vectors** based on co-occurrence information (e.g., adding to the memory vector of w the signal vectors of context words) or by training a neural-network[52] language model in the task of **predicting co-occurrence** data (e.g., given the signal vector of w, predict w_{+1}). The review presented in the following sections adopts the terminology established in Baroni, Dinu, & Kruszewski (2014): DSMs based on accumulation of co-occurrence counts are referred to as **count** DSMs, while those based on neural networks architectures are referred to as **predict** DSMs.

In the following sections, we describe two count DSMs based on accumulation of signal vectors: **BEAGLE** (acronym for *Bound Encoding of the AGgregate Linguistic*

[51] In this work, we adopt the terminological opposition between *signal* and *memory* vectors established by Recchia et al. (2010). In the following sections, when describing the features of specific DSM implementations, we will specify the terminological mapping between the general concepts of signal and memory vector and the corresponding labels.

[52] An artificial neural network (Bishop, 1995) is a computational model inspired by the architecture and the processing dynamics of the human brain. It consists of a large number of processing units (neurons) arranged in at least two layers: the input and the output layer. Neurons are linked through weighted, directed connections responsible for the spread of activation from one neuron to one or more neurons in a different layer. It is common to introduce at least one intermediate layer between the input and the output: the hidden layer. In a neural network containing three layers, the input layer activates the hidden layer, which in turn activates the output layer. Depending on the network architecture and on the training dynamics, neural networks can be seen as algorithms for classification (assign the stimulus encoded in the input layer to one or more classes, corresponding to the pattern of activation of the output layer) or for prediction (given the stimulus encoded in the input layer, predict the upcoming stimulus by producing the corresponding pattern in the output layer).

Environment), a cognitively inspired DSM designed to encode contextual as as well as sequential information (Jones & Mewhort, 2007); **Random Indexing** (Sahlgren, 2005), an incremental word space model designed to address the scalability issue affecting DSMs. We then turn to predict DSMs and describe `word2vec` (Mikolov, Chen, et al., 2013), the most prominent case of a neural-network architecture used to learn low-dimensional and high-quality vector representations from co-occurrence data.

2.4.1 Count DSMs based on signal-vectors

The DSMs reviewed in this section, BEAGLE (Jones & Mewhort, 2007) and Random Indexing (Sahlgren, 2005), can be considered the signal-based counterpart of bag-of-word DSMs discussed in section 2.3.2. The collection of co-occurrence information is based on the same general principle: updating co-occurrence counts each time a target word is encountered. While in bag-of-words DSMs counts update is implemented by incrementing the frequency values a target/feature pair is encountered within the context window, in signal-based count DSMs the update is implemented by accumulating signal vectors. As pointed out in Jones & Mewhort (2007), accumulation of random vectors is a cognitively plausible method for collecting co-occurrence information because it lets the semantic representations emerge gradually from the initial randomness; moreover, it is powerful because it allows to keep a relatively low dimensionality without using computationally expensive decomposition methods.

BEAGLE (*Bound Encoding of the AGgregate Linguistic Environment*) is a DSM designed to encode both contextual and word order information (Jones & Mewhort, 2007). BEAGLE's signal vectors (*environmental vectors*) have low dimensionality (2048 dimensions) and are dense, with values drawn from a normal distribution with mean equal to zero and standard deviation equal to the inverse of the square root of the vector dimensionality. In the publicly available implementation of BEAGLE, memory vectors for 90000 inflected target words have been extracted from the TASA corpus[53] using sentence boundaries as a defining criterion for the context window. Under the assumption that word meaning and usage can be learned together in a single pattern of vector elements (Jones & Mewhort, 2007, p. 5), BEAGLE's memory vectors (*composite lexical vectors*), are dynamically updated during training with contextual and word order information. Every time a target word t is encountered in a new sentence, its memory vector ($\vec{m}_t$) is updated by superposition with the context ($\vec{c}_t$) and the order ($\vec{o}_t$) vector: $\vec{m}_t = \vec{m}_t + \vec{c}_t + \vec{o}_t$.

Context and order vectors are calculated with different mathematical operations on the signal vectors of the words occurring in the sentence. In more detail:

- A context vector ($\vec{c}_t$, encodes which words occur with t in the sentence) is calculated by **summing** the random signal vectors ($\vec{s}_w$) of all the words in the sentence, excluding t.
  For example, in the sentence *"dogs eat bones"* the contextual vector for the target *eat* is calculated as follows: $\vec{c}_{eat} = \vec{s}_{dogs} + \vec{s}_{bones}$;

- An order vector ($\vec{o}_t$, encodes the position of t relative to the other words in the sentence) is calculated by extracting all n-grams containing t and, for each n-gram,

[53] The TASA corpus is a collection of English texts compiled by Touchstone Applied Science Associates which is considered equivalent to the amount of texts read by an average college-level student and was also used to train the Web version of LSA (Landauer & Dumais, 1997).

binding the environmental vectors of the context words using **directional circular convolution** ($*$). The order vector for a target t in a sentence is the sum of the convolutions of all the n-grams in that sentence.
Directional circular convolution (Murdock, 1982; Plate, 2003) is a method for compressing the tensor product of two vectors; this method presents a number of properties which BEAGLE fully exploits. First, it is non- commutative ($\vec{a} * \vec{b} \neq \vec{b} * \vec{a}$), and therefore appropriate to model word transitions. Second, its output encodes associative information between $\vec{a}$ and $\vec{b}$, without losing track of which features belong to $\vec{a}$ or $\vec{b}$: as a consequence, once two vectors are convoluted into a third one ($\vec{a} * \vec{b} = \vec{z}$), it is possible to apply a decoding operation (called deconvolution or correlation[54]) to get a noisy version of each vector by probing the convoluted output with the other vector (e.g., $\vec{a} \# \vec{z} \approx \vec{b}$). Third, the convoluted output $\vec{z}$ has the same dimensionality of the input vectors: this is a necessary property in the BEAGLE implementation, as order vectors need to be added to memory vectors. In our example sentence, the set of relevant n-grams (with ϕ as a static vector acting as a placeholder for t) is $\{dogs\ \phi;\ \phi\ bones;\ dogs\ \phi\ bones\}$ and the corresponding order vector is calculated as follows: $\vec{o}_{eat,dogs\ eat\ bones} = (\vec{s}_{dogs} * \phi) + (\phi * \vec{s}_{bones}) + (\vec{s}_{dogs} * \phi * \vec{s}_{bones})$.

Jones & Mewhort (2007) evaluate BEAGLE in a number of tasks and show that adding order information to the context information improves the quality of the semantic representations, and that the order information encoded in the memory vectors allows to model sentence processing data without introducing word transition rules external to the DSM.

While BEAGLE is a specific DSM, **Random Indexing** (henceforth, RI) has been developed a dimensionality reduction method (Sahlgren, 2005). In the RI approach, signal vectors (*index vectors*) are sparse and ternary (the range of the random values is restricted to -1, 1 or 0). The dimensionality of the signal vectors is independent of the size of the vocabulary and it is set as a model parameter (it usually ranges from a few hundred to a few thousands dimensions). The memory vector (*context vector*, in the RI terminology) for a target word t is accumulated by applying a context window and **summing** the signal vectors of the words in the context, which differ from BEAGLE's context vectors only in the size of the co-occurrence window and how the random signal vectors are generated. The dimensionality of RI's memory vectors is therefore identical to the dimensionality of the underlying signal vectors, and the resulting semantic representations proved robust at a dimensionality lower than that of state-of-the-art bag-of-word models (Sahlgren, 2005).

Initially developed as a method for reducing the dimensionality of the co-occurrence matrix, RI has also been further extended to encode word order information in distributional vectors. Sahlgren et al. (2008) introduce a Random Permutation Model (henceforth, RPM), based on **random permutations** (i.e., functions which take vectors as their input and produce a randomly shuffled version of them as an output). Random permutations have the same convenient properties of circular convolution (they are not commutative, allow the retrieval of a noisy approximation of their input, keep vector dimensionality), and are computationally less expensive. Sahlgren et al. (2008) report

[54]Since convolution/deconvolution operations are also at the bases of light holography (Plate, 2003), Jones & Mewhort (2007) refer to BEAGLE as a *holographic* lexicon.

that, in a number of standard tasks, RPM achieved comparable performances to BEAGLE on a subset of the Wikipedia corpus; differently from BEAGLE, RPM could also be trained the full Wikipedia corpus, with significant improvements in all tasks.

Besides the differences in implementation details (i.e., representation of signal vectors, mathematical operations employed to encode word order) BEAGLE and RI share the incrementality of their training procedure, which allows to test the quality of the semantic representations at intermediate stages of training. Incrementality in the acquisition of semantic representations is considered to be a necessary feature of a cognitively plausible model. Lack of incrementality is thus considered a limitation of bag-of-words models employing feature weighting or dimensionality reduction techniques, which apply to co-occurrence frequencies extracted from the entire corpus. In the next section we will discuss a different type of DSM which is based on signal vectors and shares with BEAGLE and RI the incrementality of the training procedure, but adopts a different training approach to learn memory vectors from co-occurrence data.

2.4.2 Predict DSMs based on signal vectors

In the previous section two DSMs were discussed, which implement count update as accumulation of signal vectors: such models can be considered the signal-vector counterpart of bag-of-words DSMs. In this section we focus on a different class of signal-based DSMs, which learn low-dimensional distributional representations by training a neural network in a supervised task: the prediction of co-occurrence data (given a target word, predict its context; given the words in the context, predict the target word).

In this work, we refer to signal-vector DSMs based on neural architectures as **predict DSMs** (Baroni, Dinu, & Kruszewski, 2014): we consider this label the most appropriate because it refers to the learning procedure (i.e., training a neural network in the task of predicting co-occurrence). In the NLP literature, predict DSMs are commonly defined as instances of **deep learning**. Here, we do not adopt this label because it is at odds with the actual architecture of the involved DSMs: in machine learning, the defining feature of a *deep* neural network is the presence of multiple hidden layers between the input and the output,[55] while predict DSMs like the one devised by Mikolov, Chen, et al. (2013) and widely employed in the NLP community have only one hidden layer.

In the NLP literature, the most prominent predict DSMs is `word2vec` (Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013), a three-layer neural network trained in two complementary versions of a co-occurrence prediction task (figure 2.12). In the first version of the task, which corresponds to the *continuous bag-of-words* model (*cbow*), the network is presented with the sum of the vectors of the words occurring in the context window (e.g., two words to the right and to the left of the target) and it has to predict (i.e., activate in the output layer) the vector representation of the target; in the second version of the task, corresponding to the *skip-gram* model, the network is presented with the representation of the target word and it predicts the vectors of the context words occurring in the given window. The pattern of activation of input and output layers is a highly sparse one-hot **signal vector** (only

[55] Multiple hidden layers are used to model information at increasing levels of abstraction: for example, in an image processing task, a first hidden layer can be used to identify edges based on the information contained in the input layer (e.g., a vector containing intensity values for each pixel in an image); a second hidden layer is employed to identify shapes on the basis of the edge information encoded in the first hidden layer, and so on.

one vector position is set to one, and all the others are left to zero); as a consequence, `word2vec`'s signal vectors are orthogonal and have a very high dimensionality, equal to the size of the vocabulary.

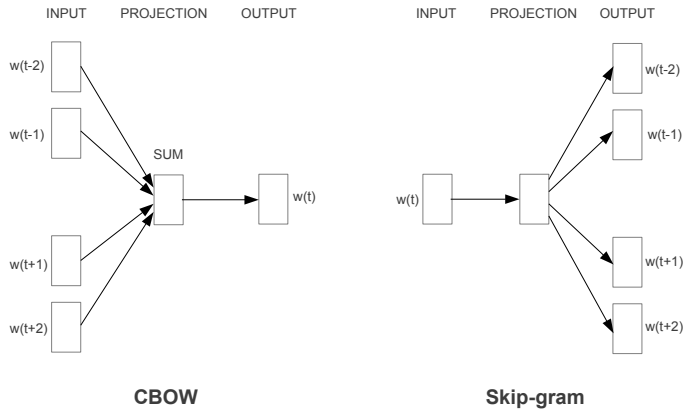

Figure 2.12: Continuous bag-of-words vs. skip-gram model (Mikolov, Chen, et al., 2013, p. 5)

.

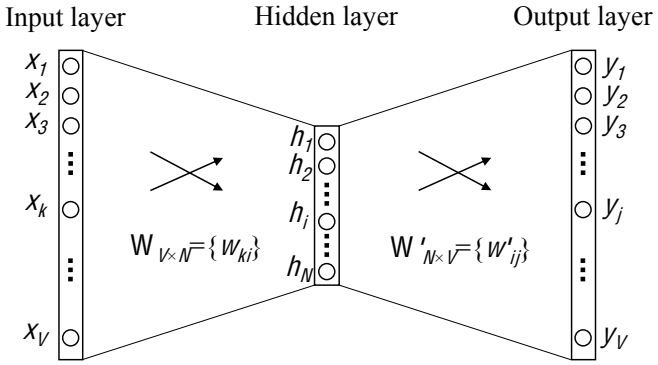

Figure 2.13: Continuous bag-of-words model: bigram version (Rong, 2014, p. 2)

.

For reasons of clarity and space, we follow Rong (2014) and illustrate here a bigram implementation of the continuous bag-of-words version of `word2vec`. In this implementation of the *cbow* model, the network is presented with the vector representation of a word w (the context) and activates in the output layer the vector representation of the upcoming word (the target). Figure 2.13 illustrates the architecture of the network: V is the vocabulary size, N is the dimensionality of the hidden layer, set to a value significantly smaller than V. Every neuron in the input layer $\{x_1...x_V\}$ is connected with every neuron in the hidden layer $\{h_1...h_N\}$. In turn, every neuron in the hidden layer is connected to every neuron in the output layer $\{y_1...y_V\}$. The weights of the connections

are initialized randomly (in the `word2vec` implementation with values ranging from -0.5 to 0.5) and the corresponding vectors are unique. The weighted connections between neurons in different layers form two matrices: W, which connects input and hidden layer and has shape $V \times N$; W', which connects hidden and output layer and has shape $N \times V$. When the network is presented with the context word w, whose one-hot signal vector contains a 1 at the kth position, the activation propagates from x_k to h_x to y_j. The one-hot signal vector containing a 1 at the jth position is the network's prediction for the word following w.

Scanning the corpus with a sliding window of the desired size, the network is trained, adjusting the weights in W and W'. The weights in the matrices are estimated in supervised fashion: they are set to maximize the probability of predicting the output (the target word, in *cbow*; the corpus-observed context for *skip-gram*[56]) given the input. Once the network has been trained with either of the methods (*skip-gram* or *cbow*), dense and low dimensional **memory vectors** are drawn from the matrix of weights W and can be used to quantify similarity. The dimensionality of the weight matrix (and as a consequence, of the memory vectors) can be set as a model parameter to values that are clearly lower than the vocabulary size. Because of the condensed semantic representation they encode, `word2vec`'s memory vectors are also referred to as **embeddings**.[57]

Mikolov, Chen, et al. (2013) show that the memory vectors produced by `word2vec` are suitable to model different types of relationships between words, addressing the issue of a better characterization of semantic similarity in DSMs discussed in section 2.1.3.2. The semantic relation holding between two words (e.g., *king* and *man*) can be characterized in terms of a shift vector by subtracting the vector of the second word from the vector of the first one ($\vec{v}_{king} - \vec{v}_{men}$). The shift vector can then be added another word (e.g., *woman*) to solve analogical relations: $king : men = x : woman$ is implemented as $\vec{v}_{king} - \vec{v}_{men} + \vec{v}_{woman} \approx \vec{v}_{queen}$. In comparative evaluation studies on semantic similarity and analogy tasks, `word2vec` achieved better performances than term-term (Baroni, Bernardi, & Zamparelli, 2014) or term-document (Mikolov, Sutskever, et al., 2013) DSMs. Such evaluation studies, however, targeted a very specific implementation of LSA (Mikolov, Sutskever, et al., 2013) and a relatively restricted number of parameter configurations for the involved bag-of-words models (Baroni, Dinu, & Kruszewski, 2014). An additional contribution to the comparison between bag-of-words DSMs and `word2vec` comes from Levy & Goldberg (2014b), who demonstrated that the skip-gram model is implicitly factorizing a term-term matrix whose cells contain PPMI scores shifted by a global constant (see discussion of shifted PPMI in section 2.3.3.1).

[56] In the *skip-gram* implementation, the network is trained relying on both positive and negative evidence. The weights are adjusted to maximize the similarity between predicted and observed output (positive sample), and to minimize the similarity between the predicted output and a number of context words that did not co-occur with the target (negative sample). Negative sampling involves a set of training parameters (size of the negative sample, frequency criteria for selecting candidates of the sample) that are not discussed here for reasons of space and and because this work is not concerned with `word2vec`.

[57] The label "embedding" is to be shared with the output of any dimensionality reduction method which projects a sparse, high dimensional representation into a dense, reduced representation (e.g., SVD, NMF).

2.4.3 Signal-vectors vs. co-occurrence based DSMs

In the previous sections, three DSM implementations (BEAGLE, RI, and `word2vec`) were described, which learn low-dimensional semantic representations (memory vectors) on the basis of randomly generated vectors employed as word signatures (signal vectors). In this section, we elaborate on the comparison between those models and bag-of-words DSMs and motivate the choice of keeping the focus of this thesis on the latter.

Let us start from the status of signal-vector based DSMs as dimensionality reduction methods. Differently from bag-of-words DSMs and, to a lesser extent, SVD or NMF (see 2.3.3.2) the dimensions of RI and `word2vec`[58] are opaque and not interpretable. A possible strategy for the characterization of the semantic representations encoded in the embeddings is the extraction of the nearest neighbors for a set of target words; word neighborhoods can be qualitatively analyzed to infer the meaning components dominating the semantic representations. Within the `word2vec` framework, an alternative based on a skip-gram model trained on parsed data is proposed by Levy & Goldberg (2014a). In this case, the input of the network is the one-hot representation of the target (e.g., *dog*) and the output layer encodes dependency-typed contexts (e.g., *subject-bark*, *coordination-cat*). Levy & Goldberg (2014a) identify the most salient dimensions for a target word by presenting the network with the target's one-hot vector, and letting the activation spread to the output layer: the most activated dimensions in the output layer are considered the most salient meaning components associated to the target. Albeit interesting, especially for the introduction of dependency-based contexts, this solution rests on the qualitative difference between the input and the output but it does not address the issue of interpreting the *dimensions* of the embeddings.

We now turn to the comparison of bag-of-words DSMs with BEAGLE and RI with respect to incrementality and word order encoding. Neither of those features are at odds with a bag-of-words implementation. As a matter of fact, bag-of-words DSMs are extracted in a completely incremental fashion: a co-occurrence matrix containing raw frequency counts is a full-fledged DSM, and the collection procedure is incremental (and equivalent to a RI model with one-hot vectors as signal vectors), and testable at intermediate stages of training.[59] The crucial difference between bag-of-words and RI in their approach to the collection of co-occurrences rests in their memory consumption: updating a co-occurrence matrix containing many thousands of dimensions (bag-of-words DSM) is computationally way more costly than summing vectors with hundreds, or at most few thousands, dimensions (RI). As far as word order encoding is concerned, in section 2.3.2.1 we described the different ways in which the position of a feature word relative to the target can be encoded in a bag-of-words DSM. Even if a co-occurrence matrix containing fine-grained positional features (position and distance of the context relative to the target, e.g., two words to the right) is not equivalent to the complex n-gram representation encoded in BEAGLE or RPM, it can be employed to estimate both semantic similarity and word transitions.

Despite the popularity of `word2vec`, bag-of-words models are still widely employed if we consider the whole Computational Linguistics field and not only the NLP community; this is only partially due to the fact that bag-of-words DSM can be built easily

[58] The arguments put forward for RI can also be extended to BEAGLE. We do not list BEAGLE here because it is a specific DSM implementation and not a dimensionality reduction method.

[59] Moreover, the incrementality of signal-vector models employing stopword lists based on frequency counts is questionable.

from co-occurrence data, without resorting to off-the-shelf packages like `word2vec`. An additional reason is the lack of interpretable dimensions, a crucial limitation when dealing with tasks or applications which rely on interpretable distributional features (e.g., the computation of inclusion measures for hypernym/hyponym identification described in 2.1.3.2). An additional advantageous feature of the (count) bag-of-words approach to co-occurrence is that, after one pass on the entire corpus, frequency information from a bag-of-words matrix can be further manipulated, for example aggregating counts by applying basis mapping (e.g., converting fine grained positional information into a symmetric context window).[60]

To conclude, a very pragmatic reason determined our choice of keeping the evaluation focus on bag-of-words DSMs: when `word2vec` became popular, the project described in this thesis had already been defined and experiments were running. Large-scale evaluation work has to face concrete limitations (it is simply not possible to evaluate everything) and count DSMs already provide a fairly good overview/understanding of a sensible space of meta-parameters; the same does not hold for for neural embeddings, where new architectures (often as slight variations on previous models) keep on being devised, making the identification of a clear-cut parameter space particularly difficult.

[60] In its standard implementation as a dimensionality reduction technique, RI applies to the co-occurrence information collected in a bag-of-words DSM.

3

Evaluation of DSMs

The previous chapter characterized Distributional Semantic Models in terms of their cognitive and linguistic motivations, as well as their formal properties and implementation details; the potential limitations of DSMs as full-fledged models of semantics have also been pointed out: as discussed in section 2.1.3.2, distributional representations may be considered unsatisfactory because the type of similarity they encode is too broad. This issue is aptly described by Sahlgren (2008, p. 37): "The distributional hypothesis, as motivated by the works of Zellig Harris, is a strong methodological claim with a weak semantic foundation. It states that *differences* of meaning correlate with *differences* of distribution, but it neither specifies *what kind* of distributional information we should look for, nor *what kind* of meaning differences it mediates." This chapter addresses the evaluation of DSMs, which brings together the two sides of the methodological issue raised by Sahlgren: what kind of meaning differences distributional information can capture and what kind of distributional information is necessary for that. Instead of operating on the level of general similarity that is supposed to be mirrored by some general distribution, DSM evaluation narrows down the broadness of the distributional hypothesis by comparing the performance of specific distributional information (in the form of specific DSM configurations) against the manifestation of specific types of similarity (encoded in the form of various similarity tasks). An **evaluation task** can be interpreted as the formulation of a hypothesis on the nature of the DSM representations in terms of a specific **experimental setting**: for example, the hypothesis that similarity between word distributional representations is akin to synonymy can be tested in a multiple choice task based on datasets like TOEFL (Landauer & Dumais, 1997) or ESL (Turney, 2001). The definition of an evaluation task also includes the appropriate **performance** measure(s), for example accuracy in a multiple choice task or correlation in the task of modeling similarity judgments. Note that while tasks are defined for specific datasets, the definition of an evaluation task is in principle independent of the employed dataset; it is the dataset which contributes the "type of similarity" tested in specific experiments (i.e., synonymy, antonymy, topical relatedness, etc.).

The aim of this chapter is to provide a taxonomy of the tasks employed for the evaluation of a DSM, as well as an overview of the state of the art for the tasks in the focus of this dissertation. There are a number of classification dimensions according to which a taxonomy of DSM evaluation tasks can be structured; in section 3.1 we discuss these classification dimensions and motivate the criteria according to which the reminder of the chapter will be structured. Section 3.2 will discuss semantic similarity

tasks and section 3.3 will summarize the literature on DSM evaluation on cognitive modeling tasks. Finally, section 3.4 will describe an evaluation task devised for the purpose of this dissertation, namely multiple choice on semantic priming datasets, and will provide more details on the corresponding datasets.

3.1 Classification criteria

Let us start with the distinction between **intrinsic and extrinsic evaluation**, which modulates the immediacy between the distributional representation and the quantification of DSM performance. A task is intrinsic when the semantic representations produced by the evaluated DSM are directly tested on their capability of capturing the type of similarity (e.g, synonymy, antonymy, etc.) encoded in the selected dataset; there are no intermediate steps between the computation of the DSM representations and their evaluation, therefore the variation in performance is exclusively determined by the manipulation of the DSM parameters. On the other hand, extrinsic evaluation of DSMs targets the impact of different vector representations on the performance of NLP systems of which DSM is only a pipeline component, e.g., a machine translation system; performance is measured according to the criteria of the task at issue (e.g., BLEU score e.g., (Papineni et al., 2002) for translation quality), and variation in performance cannot be ascribed to the DSM parameters only, as it is likely to be determined by the interaction between specific DSM features and features of other components of the pipeline. Note that the majority of the DSM applications described in section 2.1.3 (e.g., word-sense disambiguation, modeling of semantic compositionality) are still to be classified under the label of intrinsic evaluation: despite the more complex operations performed on DSM vectors, the output of such operations is still employed for a direct quantification of similarity. Given that the goal of this thesis is to achieve a thorough understanding of the effects of different DSM parameters on the performance of the models, we will focus on intrinsic tasks and turn to extrinsic evaluation only in the end of chapter 8.

A further criterion for the classification of DSM evaluation tasks concerns the design of the evaluation setting, and it contrasts tasks targeting similarity between **single words** from tasks targeting similarity between **word pairs**. Similarity between single words (e.g., *dog* and *wolf*) is quantified as the degree of correspondence between the properties of the target items. Standard semantic similarity tasks such as the multiple choice synonymy test or prediction or similarity judgments are examples of this type of task. Similarity between word pairs (e.g., *mason:stone* and *carpenter:wood*) is quantified as the degree of correspondence between the relations holding between the members of one pair and the relations holding between the members of the other pair. Examples of this type of task are modeling of analogies (Turney, 2008; Mikolov, Chen, et al., 2013; Mikolov, Wen-tau, & Zweig, 2013; Mikolov, Sutskever, et al., 2013) as well as supervised classification of semantic relations. Note that the distinction between attributional and relational similarity introduced by Turney (2006) conflates task design (single words vs. word pairs) with the nature of the modeled relations. It is, however, desirable to keep the two classification criteria separate and let specific evaluation datasets determine which of the many "similarities" is targeted in a specific modeling experiment. Evaluation tasks of either design can target any type of semantic relation: paradigmatic (synonymy, antonymy, hypernymy, etc.) or syntagmatic (typical actions, patients, in-

struments, etc.). This thesis focuses on similarity between single words, mainly because of practical reasons: a thorough understanding of the parameter space governing word similarity is a necessary first step before moving on to similarity between word pairs; the extension of the evaluation methodology proposed in this thesis to analogy or supervised classification of semantic relations is left for future work.

A third criterion for the classification for DSM evaluation concerns the type of the modeled data, and distinguishes tasks based on **semantic similarity tests** from **cognitive modeling** tasks based on behavioral data. All evaluation tasks which are usually referred to as "standard" fall into the first category: multiple choice, word similarity, or analogy datasets are lists of word pairs which stand to each other in a specific "type of similarity"; DSMs are evaluated in their capacity of solving the test in the same way a human subject would. By contrast, cognitive modeling tasks evaluate DSMs in their capacity of predicting speakers' behavior in psycholinguistic experiments (which have in turn been designed to test specific types of similarity). In practice, the starting point of this type of evaluation is identical to that of the standard tasks: a list of words or word pairs selected according to certain criteria; crucially, what is tested in this case is not whether specific DSM representations can account for the criteria which led to the collection of the experimental items (i.e., the experimenter's hypothesis), but whether they account for the actual experimental result (i.e., human behavior).

3.2 Semantic similarity tests

In this section, we provide an overview to semantic similarity tests for DSMs, which are considered standard in DSM evaluation. Note that this list is not an exhaustive one: in fact, we keep our focus on tasks and datasets which are relevant for this thesis, with some exceptions. We therefore leave aside tasks based on attributional similarity (e.g., analogy prediction), as well as tasks targeting sentence-level similarity. Classification of semantic relations as well as more linguistically informed tasks such as modeling of entailment, reference, and compositionality have been summarized in section 2.1.3.

3.2.1 Multiple-choice (synonymy) test

In a multiple-choice task, the DSM is presented with a target word (e.g., *levied*) and a set of candidates for a specific semantic relation, usually synonymy (e.g., *imposed, believed, requested, correlated*). The task for the DSM is to choose the best candidate (in our running example, *imposed*). Performance is quantified in terms of **accuracy**.

Datasets The most employed datasets for multiple-choice evaluation are TOEFL, acronym of Test Of English as a Foreign Language, introduced by Landauer & Dumais (1997) and ESL, acronym of English as a Second Language, introduced Turney (2001). TOEFL contains 80 synonym questions composed by a target word and four candidates. ESL contains 50 synonym questions with 4 choices per question and a sentence context in which the correct answer needs to fit.

State of the Art The state of the art performance for TOEFL, 100%, is by Bullinaria & Levy (2012); it was achieved by employing a window-based DSM trained on UkWaC, reduced with PCA and with manipulation of Caron p (cf. section 2.3.3.2). As for ESL,

the state of the art of corpus-based systems, 82%, is by Terra & Clarke (2003), who in turn implement the PMI-IR algorithm by Turney (2001).

3.2.2 Prediction of similarity ratings

The prediction of human similarity ratings is implemented in terms of a mathematical comparison between word similarity (or relatedness) as predicted by a DSM and the similarity (or relatedness) judgments produced by human annotators. In this task, DSM performance is quantified in terms of Spearman or Pearson correlation. Note that we decided to list the modeling of similarity judgments among the standard tasks: such choice (which can be questioned) is determined by the consideration that while experimental subjects of priming, reading, EEG, or free association norms are not instructed to focus on a specific "type of similarity", this is the case for the subjects involved in the collection of similarity/relatedness judgments.

Datasets The classic dataset for correlation to human ratings is the Rubenstein & Goodenough (RG65) dataset, which contains 65 items rated for similarity on a scale between 0 and 4 (Rubenstein & Goodenough, 1965). Next, comes the WordSim-353 dataset (WS353), which contains 353 noun pairs rated on a scale between 0 and 10 (Finkelstein et al., 2002). WordSim-353 comprises two subsets and it contains ratings for both similarity and relatedness. A more recent dataset is MEN, introduced by Bruni et al. (2013): it contains 3000 word pairs annotated on a 50 point scale as a result of a crowdsourcing experiment.

State of the Art The state of the art on RG65 is 0.86 *rho*, achieved by Hassan & Mihalcea (2011) by using a hybrid approach (word sense disambiguation and knowledge-base linking). State of the art on WS353 is 0.81 *rho*, held by the hybrid model by Halawi et al. (2012) who augmented a neural model with WordNet information. Finally, state of the art on MEN, 0.80 *rho*, has been achieved by the word2vec embeddings in the comparative evaluation by Baroni, Dinu, & Kruszewski (2014).

3.2.3 Clustering

Distributional relatedness between words is used to assign them to certain pre-defined semantic classes. From a lexical semantic point of view, the targeted relation is co-hyponymy. Performance in this task is quantified in terms of purity and entropy.

Datasets The clustering datasets evaluated in Baroni & Lenci (2010) represent the reference for DSM evaluation in this task. Their selection includes the Almuhareb-Poesio set (henceforth, AP), containing 402 nouns grouped into 21 classes (Almuhareb, 2006); the Battig set (henceforth, BATTIG), containing 83 concrete nouns grouped into 10 classes (Van Overschelde et al., 2004); the ESSLLI 2008 set (henceforth, ESSLLI), containing 44 concrete nouns grouped into 6 classes;[1] and the Mitchell set (henceforth, MITCHELL), containing 60 nouns grouped into 12 classes (T. Mitchell et al., 2008).

[1] `http://wordspace.collocations.de/doku.php/data:esslli2008:concrete_nouns_categorization`

State of the Art The state of the art for AP corresponds to a purity of 0.79, achieved by the syntax-based model by Rothenhäusler & Schütze (2009). As for BATTIG, best performance is held by the neural embeddings in Baroni, Dinu, & Kruszewski (2014). State of the art for ESSLLI is held by the approach of Katrenko & Adriaans (2008), which consists in the application of manually defined patterns to a large web corpus. Finally, the SVD-tuned window-based DSM by Bullinaria & Levy (2012) holds the state of the art for the MITCHELL dataset, with a purity of 0.94.

3.3 Cognitive modeling

Cognitive modeling evaluation is challenging from many points of view. First, the modeled data are typically continuous (reaction times, EEG signal) or, when categorical (free association norms), much less constrained compared to the data used in standard tasks. Second, behavioral data may just not have confirmed the experimenter's hypothesis, but a DSM could still find an effect. In this case, interpretation of DSM modeling results becomes particularly tricky: is it that the DSM is wrong and is overpredicting effects, or is it indeed capturing an additional real effect which was not targeted in the experimental setting? Third, speaker's behavior is the result of the interplay of many factors which, in the case of continuous data, get accounted for by applying regression analysis settings. Such settings employ a rich set of predictors (e.g., random effects for subject or items, frequency effects, etc.), which already account for a lot of variance in the data. Therefore, it is not straightforward that the DSM can provide an additional interpretation level, because it is possible that a part of its potential contribution is already accounted for by some other predictors. To sum up, while quantifying performance of DSMs is straightforward in standard tasks, it is less so in the case of cognitive modeling tasks.

In the following sections, we will elaborate on two cognitive tasks which will be in the experimental focus of this dissertation: the modeling of free association norms and of that of priming datasets. Once again, this is not an exhaustive selection: the modeling of brain data, for example, falls out of the scope of this dissertation. For an overview, we refer the reader to Bullinaria & Levy (2013) and references therein.

3.3.1 Prediction of free association norms

A free association for a stimulus word is "the first word that comes to your mind when you hear...". Free associations are considered as a cue into the organization of the mental lexicon. There has always been debate, however, concerning the nature of the cognitive processes regulating free associations: while earlier theories considered free associations as the result of learning by contiguity (James, 1890), later theories have accounted for them in terms of symbolic processes and complex semantic structures (H. Clark, 1970); empirical annotation contributed to the characterization of free associates as a mixture of syntagmatic relations, which hold between contiguous word, and paradigmatic relations, which hold between semantically related words (Brown & Berko, 1960; Fitzpatrick, 2007).

Free association datasets represent an excellent starting point for DSM modeling, as they are cognitively motivated (hence semantically plausible) and usually large, and thus allow for robust evaluation and for a better selection of the distractors.

Based on a free association dataset, is it possible to set up two types of evaluation tasks: in a regular free association task, the corpus-based model needs to generate the response for a specific stimulus (*cat*→? *significant*→?); in the reverse free association task, the model is shown a number of response words and needs to guess the corresponding stimulus (? → *away, minded, gone, present, ill*). A problem with such evaluation setups is the presence of an unrestricted set of possible responses in combination with a discrete association task, which requires the algorithm to pick exactly the right answer out of tens of thousands of possible responses. This feature makes this task much more difficult than the multiple-choice tasks often used to evaluate distributional semantic models. Additionally, free association datasets have also been employed in a classification setting, in which given a stimulus word and a series of candidate responses, the corpus-based model needs to identify the most frequent response to the stimulus (i.e., the word that was produced by the highest number of subjects as a response).

Datasets The largest free-association datasets are the Edinburgh Associative Thesaurus (EAT, 8210 stimuli, 100 subjects) by Kiss et al. (1973) and the University of South Florida Free Association Norms (USF, 5019 stimuli, 6000 subjects) by Nelson et al. (2004).

State of the art A task derived from the EAT norms was used in the ESSLLI 2008 shared task.[2] Results from first-order co-occurrence data (collocations) turned out to be much better than those from second-order DSMs (vector similarity), in line with previous findings by Rapp (2002) and Wettler et al. (2005).

A similar picture emerges from studies on the (reverse) multiword association task. Models based on first-order co-occurrence outperform models based on vector similarity. This superiority, however, has not been validated via a direct comparison: results were obtained by studies with different features and goals (see Rapp (2014) for a review; see Griffiths et al. (2007) for an evaluation of models based on Latent Semantic Analysis). A specific feature of successful studies on the multiword association task is that they introduce an element of directionality (Rapp, 2013, 2014), which allows a correct implementation of the directionality of the modeled effects (from stimulus to response).

The CogALex shared task 2014 (Rapp & Zock, 2014) has proposed a reverse multiword association task based on a set of 2000 stimuli from EAT. The task was very challenging: the winning system, which used first-order statistics to re-rank the output of a "standard" DSM, only achieved 35% accuracy (Ghosh et al., 2014). In chapter 8 we will discuss our contribution to the CogALex shared task (Lapesa & Evert, 2014b).

3.3.2 Priming: modeling of reaction times

Priming datasets contain collections of word triples: a target word (e.g., *dog*); a consistent prime, i.e., a word standing in a specific semantic relation to the target (e.g., *cat* for the cohyponymy relation); and an inconsistent prime, i.e., a word that is unrelated to the target (e.g., *stone*). For each pair of target and prime, priming datasets also list the average RTs or, in the fortunate cases, the full set of measurements (with data per subject). Note that experimental items from priming experiments can be used in

[2] `http://wordspace.collocations.de/doku.php/data:esslli2008:correlation_with_free_association_norms`

a categorical setting (e.g., multiple choice task presented in chapter 8), more akin to standard tasks than to "proper" cognitive modeling: in this case, DSM evaluation can be seen as a corpus-based test of the quality of the experimental items.

Overall, the approaches to the corpus-based evaluation of priming fall into three categories:

- Pattern replication: given a set of experimental items, the similarities from the corpus-based model are expected to reproduce the pattern of experimental results. This approach is usually implemented in form of a statistical test to check for significant differences between congruent and incongruent conditions (Padó & Lapata, 2007; Hare et al., 2009).
- Correlation between DSM similarities and RTs, as in Lapesa & Evert (2013a).
- Item-based prediction of RTs or priming effect (difference between the congruent and the incongruent condition) based on DSM as well as other covariates. An example of such approach is the study by Hutchison et al. (2008) described below and the experiments by Lapesa & Evert (2013c) summarized in section 9.1.

Datasets & State of the Art In general, priming datasets have not been systematically employed in DSM evaluation. The only exception is the Hodgson dataset (Hodgson, 1991), whose priming effects have been modeled in terms of a comparison between prime-target pairs, in a pattern replication task (McDonald & Brew, 2004; Padó & Lapata, 2007; Herdağdelen et al., 2009). This dataset is, however, fairly small (143 items covering 6 relations). It is not considered in the present work because, given the number of parameters we planned to evaluate, it would have been difficult to build robust generalizations from the evaluation results.

Hutchison et al. (2008) present the result of item-based prediction of RTs on 300 target-prime pairs from the English Lexicon Project.[3] Among the employed set of predictors, distributional models are represented by LSA similarity, which turns out to be not significant. Needless to say, DSM similarity is not just LSA (and the aim of this dissertation is precisely do show how "different" DSM similarities can be): big datasets such as the Semantic Priming Project[4] or the English Lexicon Project lend themselves perfectly to large scale evaluations of DSMs on the item-based prediction of RTs.

3.4 Multiple choice on priming datasets

In this dissertation, we employ the experimental items from a number of semantic priming studies to evaluate our DSMs in a multiple choice setting.

Word triples from priming datasets represent perfect candidates for a multiple-choice task, as we can expect that distributional relatedness between the target and the consistent prime is higher than the one between the target and the inconsistent prime. Besides that, priming studies often provide large amounts of reliable experimental items, which ensures a good quality of the distractors (the inconsistent primes). The task is easier here compared to TOEFL because the choice of the best candidate is made between 2 candidates instead of 5, but the size of the datasets compensates for that. Finally, a

[3] http://elexicon.wustl.edu/

[4] http://spp.montana.edu/

note of caution is to be made concerning the interpretation of the results, which should not be taken to reflect the capability of DSMs in the modeling of the actual priming effects, but as a test of the semantic information encoded in the experimental items.

We gathered datasets from a number of priming experiments which test different types of semantic relations. We present evaluation results on six datasets:

- The first five datasets are derived from the **Semantic Priming Project** (SPP, Hutchison et al., 2013). To the best of our knowledge, our study represents the first large-scale DSM evaluation on items from this dataset. The original dataset consists of 1661 word triples collected within a large-scale project aiming at characterizing English words in terms of a set of lexical and associative/semantic characteristics, along with behavioral data from visual lexical decision and naming studies.We manually discarded all triples containing proper names, adverbs or inflected words. We then selected five subsets involving different semantic relations, namely:
  - synonyms (SYN): 436 items (e.g., *frigid–cold* as consistent prime and target);
  - antonyms (ANT): 135 items (e.g., *hot–cold*);
  - cohyponyms (COH): 159 items (e.g., *table–chair*);
  - forward phrasal associates (FPA): 144 items (e.g., *help–wanted*);
  - backward phrasal associates (BPA): 89 items (e.g., *wanted–help*).
- The sixth dataset is the **Generalized Event Knowledge** dataset (GEK). It contains a collection of 404 triples (target, consistent prime, inconsistent prime) from three priming studies conducted to demonstrate that event knowledge is responsible for facilitation of the processing of words that denote events and their participants (Ferretti et al., 2001; McRae, Hare, et al., 2005; Hare et al., 2009).[5] In more detail, the dataset contains items which test:
  - Verb-Noun priming (Ferretti et al., 2001): 118 items with 5 thematic relations, namely agent (e.g., *pay–customer*), patient (e.g., *invite–guest*), feature of the patient (e.g., *comfort–upset*), instrument (e.g., *cut–rag*), location (e.g., *confess–court*).
  - Noun-Verb priming (McRae, Hare, et al., 2005): 116 items with 4 thematic relations, namely agent (e.g., *reporter–interview*), patient (e.g., *bottle–recycle*), instrument (e.g., *chainsaw–cut*), location (e.g., *beach–tan*).
  - Noun-Noun priming (Hare et al., 2009): 170 items with 7 thematic relations, namely event-people (e.g., *trial–judge*), event-thing (e.g., *war–gun*), location-living (e.g., *gym–athlete*), location-thing (e.g., *garage–car*), people-instrument (e.g., *hiker–compass*), instrument-people (e.g., *razor–barber*), instrument-thing (e.g., *scissors–hair*).

[5] The GEK dataset has already been evaluated in Lapesa & Evert (2013a,b,c). These studies should be considered as psycholinguistically oriented pilots with respect to the work presented in this thesis. They adopted a different (more restricted) set of DSM parameters and evaluation tasks (multiple choice classification on experimental items, correlation between distributional relatedness and reaction times, item-based prediction of reaction times based on distributional information), and take a finer-grained perspective on the GEK dataset (which is analyzed per subset and per thematic relation). For a summary of the findings of these studies, refer to section 9.1.

4

Experimental setting

When it comes to DSM evaluation, there are always more parameters and tasks to explore. Evaluation studies, however, do face practical issues and the choice of parameters (e.g., window size) and parameter values (e.g., one, ten, twenty words) is determined by the interplay of many factors, for example: practical considerations concerning what is computationally feasible; the state of the art in the field and the designer's feeling of what is yet to be explored or deserves a more thorough exploration; the nature of the datasets to be modeled. The computational demands of the selected evaluation methodology also determine the shape of the parameter space; we will elaborate more on the property of different evaluation methods in chapter 5, but we anticipate here that they do differ significantly in their computational costs depending on whether they require a fully factorial design (i.e, carrying out experiments with all parameter value combinations) or approach evaluation incrementally (i.e., given a set of parameters {a, b, c} find the best value for a, set it, proceed to b, and then to c).

The aim of this chapter is to spell out the experimental setting of the evaluation studies presented in chapter 6 (term-term DSMs) and 7 (dependency-based DSMs), and further explored in chapter 8. The structure of the chapter is as follows. Sections 4.1-4.4 describe the design choices concerning parameters and parameter values involved in the **DSM extraction, manipulation, and quantification of similarity**. The structure of sections 4.1-4.4 mirrors that of section 2.3, in which DSMs parameters are characterized both theoretically and in terms of their most explored values in comparable large-scale studies. Section 4.6 focusses on the implementation details concerning the extraction and evaluation of the DSMs, and it describes the **computational tools** employed for the experiments. Building on the introduction to evaluation tasks and datasets presented in chapter 3, section 4.5 outlines the **selection of tasks and datasets**, the choice of the measure for DSM performance, as well as the task-specific tools employed. Section 4.7 concludes the chapter.

4.1 Corpus selection and pre-processing

In the experiments reported in this thesis, DSMs were built from the following selection of corpora (refer to section 2.3.1 for a detailed description of the corpora):

1. **Source corpus**: British National Corpus[1], WaCkypedia_EN (2009 dump) and

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

ukWaC;[2]

Window-based models were built relying on the part-of-speech and lemma annotation available with the original distribution of the corpora. For the dependency-based experiments, pre-processing involved the manipulation of two further parameters:

1. **Annotation pipeline** (part-of-speech tagging and dependency parsing): Tree-Tagger (Schmid, 1995) and MALT parser (Nivre, 2003)[3]; Stanford CoreNLP (version `3.5.1`), bidirectional part-of-speech tagger and Neural Network parser (Chen & Manning, 2014);

2. **Format of the (Stanford) dependency relations**: Basic vs. collapsed with propagation of conjuncts (De Marneffe et al., 2006; De Marneffe & Manning, 2008).

In what follows, we provide more details concerning these parameters and discuss the choices made with respect to their values.

Format of the Dependency Relations Dependency relations and their labels represent the building blocks of dependency paths, which are in turn the context-defining criterion in the extraction of co-occurrences for dependency-based DSMs. We selected the Stanford typed dependencies (De Marneffe et al., 2006; De Marneffe & Manning, 2008) for reasons of replicability and comparability (they are widely used in NLP applications) and because of the richness of dependency styles available.[4] Stanford typed dependencies represent grammatical relationships in a sentence as triples of relations between pairs of words (e.g., "dog" is the subject of "eat"), and they map directly into a directed graph representation in which words are nodes and dependency relations are labelled edges.

Stanford typed dependencies allow the user to choose among different styles for the representation of the dependencies, ranging from more surface-oriented representations to more lexicalized semantically interpreted ones.

The most surface-oriented style of Stanford dependencies is the `basic` representation, in which each token corresponds to a node in the dependency graph. An example of a dependency graph labelled with Stanford basic dependencies is shown in figure 4.1:[5]

[2]Both ukWaC and WaCkypedia_EN are available from `http://wacky.sslmit.unibo.it/doku.php?id=corpora`.

[3]The original distributions of Wackypedia_EN and UkWaC are already annotated with Tree-Tagger and MALT parser (version 1.8.1). Since annotation with those tools is not publicly available for the BNC and we wanted to rely on the very same annotation pipeline (same versions of tagger and parser, same models), we decided to repeat the tagging and parsing process also for the other two corpora.

[4]Starting from version `3.5.2` (end of April, 2014), Stanford CoreNLP switched to Universal Dependencies (`http://universaldependencies.org/`) as a default representation for the output of parser (Stanford Dependencies are still available). Universal Dependencies (UDs) aim at providing a language-independent inventory of grammatical categories which to facilitate cross-lingual comparison. Since then, further work on the Stanford parsers has focussed on UDs; in particular, two further versions of UDs have been developed (*enhanced* and *enhanced++*), which extend the *basic* UD format in a way (roughly) equivalent way to the way *collapsed* and *CCProcessed* extend the standard dependencies, which are adopted in this thesis. For more details on the *enhanced/enhanced++* UDs and a comparison to the *collapsed/CCprocessed* format, refer to Schuster & Manning (2016).

[5]For a detailed description of the dependency relations, refer to the Stanford Typed Dependencies manual (De Marneffe & Manning, 2008).

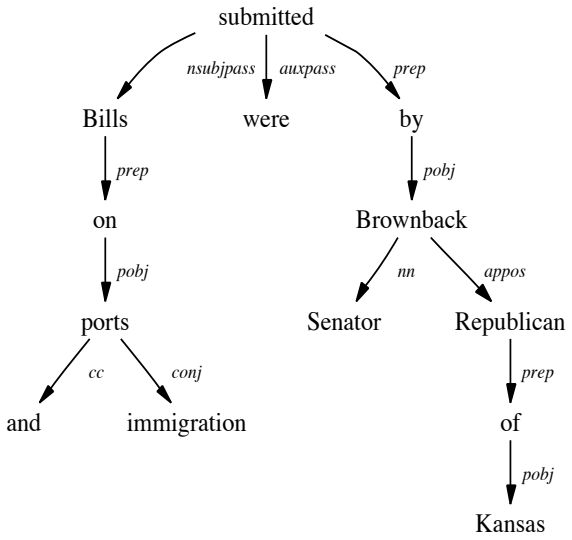

Figure 4.1: Stanford typed dependencies - Basic representation - Dependency graph for the sentence: "Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas" (De Marneffe et al., 2006, p. 2)

As an alternative to the basic variant, Stanford type dependencies provide a *collapsed* format, a more semantically interpreted representation aimed at simplifying the patterns for relation extraction. In the collapsed representation:

- Dependencies involving prepositions and conjunctions are collapsed to direct dependencies between content words;
- Multi-word constructions functioning as prepositions (e.g., "because of", "on behalf of") are collapsed as a single dependency link.

Figure 4.2 shows the collapsed dependency graph for the sentence in figure 4.1. A comparison between the two dependency graphs shows how preposition collapsing turned the two-step dependency (`prep`, preposition, and `pobj`, prepositional object) connecting "Bills", "on", and "ports" into a one-step dependency, explicitly encoding the preposition in the dependency label: `prep_on`. The same collapsing procedure also affected "submitted by Brownback" (`prep_by`) and "Republican of Kansas" (`prep_of`). Figure 4.2 also shows how the conjunction relation (`conj`) between "ports" and "immigration" gets updated to a lexicalized conjunction relation (`conj_and`).

Another difference between the basic and collapsed representation is the treatment of relative clauses. The comparison between figure 4.3 and 4.4 shows how the information concerning the referent of a relative clause is exploited to introduce a direct relation between the verb of the relative clause and the noun heading the relative clause, introducing a subject (`nsubj`) relation between "man" and "love".

Stanford typed dependencies also offer the possibility of propagating dependencies involving conjuncts: this representation builds upon the `collapsed` one, and is referred to as `CCprocessed`. Figure 4.5 shows the collapsed and propagated dependency graph

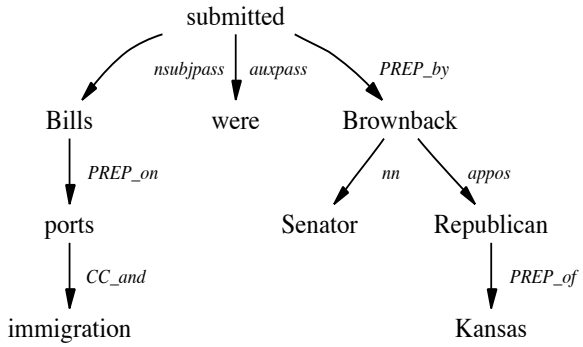

Figure 4.2: Stanford typed dependencies - Collapsed representation - Dependency graph for the sentence: "Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas" (De Marneffe et al., 2006, p. 3)

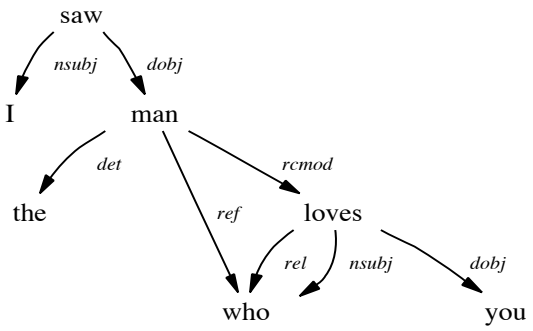

Figure 4.3: Stanford typed dependencies - Basic representation - Dependency graph for the sentence: "I saw the man who loves you" (De Marneffe et al., 2006, p. 3)

for the sentence in figure 4.2. The conjunction relation between "ports" and "immigration" was exploited to introduce a prepositional relation (`prep_on`) between "Bills" and "immigration" (in addition to the original one between "Bills" and "ports"). In addition, the `prep_by` relation between "submitted" and "Brownback" received further semantic interpretation thanks to the identification of a passive construction, and it was turned into `agent`.

The brief survey of Stanford typed dependencies conducted in this section should make it clear why we considered them as the best choice in terms of dependency relation schema for the extraction of syntax-based DSMs. We already discussed in section 2.3.2.3 how the degree of lexicalization and semantic interpretation of the dependency graph represents a key point in the definition of context paths for dependency-based DSMs: the availability of different Stanford dependency formats provided us with a robust, well-established and fully replicable strategy. For our experiments, we extracted dependency-based co-occurrences based on the *basic* (surface-oriented, one node per token in the dependency graph) and *CCprocessed* (collapsed, lexicalised dependency

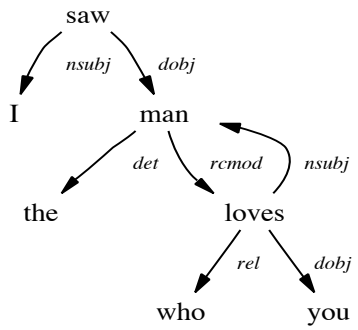

Figure 4.4: Stanford typed dependencies - Collapsed representation - Dependency graph for the sentence: "I saw the man who loves you" (De Marneffe et al., 2006, p. 3)

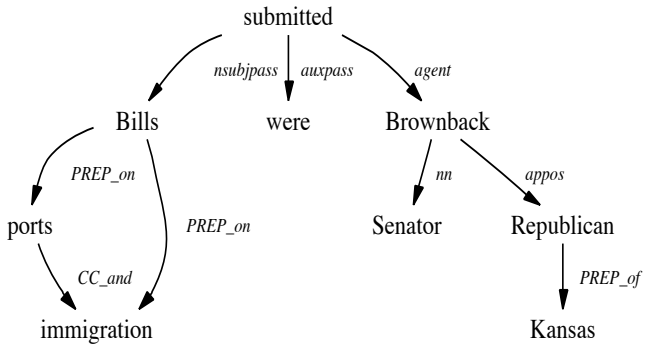

Figure 4.5: Stanford typed dependencies - CCprocessed representation - Dependency graph for the sentence: "Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas" (De Marneffe et al., 2006, p. 4)

labels with propagation of conjuncts) format. We considered the *collapsed* format too similar to the *CCprocessed* one to be worth introducing an additional parameter in our experimental setting.

Annotation Pipeline: Part-of-speech Tagger and Dependency Parser To the best of our knowledge, no study has been yet conducted to evaluate the impact of the use of different annotation pipelines on the performance of DSMs. Comparable large scale studies rely only on one parser: Minipar[6] in (Padó & Lapata, 2007) Dependency Vectors, or a pipeline composed by Tree-Tagger (Schmid, 1995) and MALT parser (Nivre, 2003) in (Baroni & Lenci, 2010) Distributional Memory.

Besides the fact that we were interested in using Stanford Dependencies for all the reasons described in the previous section, our choice with respect to the annotation pipeline was also supported by the state-of-the art in dependency parsing and by the popularity of the involved annotation tools. Luckily, our desiderata over annotation

[6] `https://webdocs.cs.ualberta.ca/ lindek/minipar/`

pipelines (producing Stanford dependencies, be state-of-the-art - or near state-of-the-art - in dependency parsing, and be widely used in NLP) converged.

Based on the state-of-the-art, we selected the Stanford CoreNLP pipeline (version `3.5.1`). Within the available Stanford CoreNLP options for part-of-speech tagging and dependency parsing, we relied on a comparative evaluation of different models on the MASC corpus.[7] Table 4.1 displays (multicore) processing speed (*speed*), labelled attachment score (*LAS*) and unlabelled attachment score (*UAS*) for the different combinations POS-tagging and parsing models (when a POS model is not specified, the default option was used) available with Stanford CoreNLP `3.5.1`. The evaluation was conducted for *CCprocessed* dependencies and separately for the spoken and written data. As we were interested in robust accuracy across written and spoken data, we only report aggregated results. Results in table 4.1 identify the combination of the bidirectional tagger with the neural network dependency parser (Chen & Manning, 2014) as the most robust option. Even if a quicker option (namely, neural network parser with default POS tagger) was available, the availability of a high-performance cluster made the increased parsing time affordable.

model	speed	LAS	UAS
pcfg	508s	0.738	0.784
factored	2999s	0.744	0.794
rnn	1515s	0.752	0.800
nndep	*22s*	0.758	0.776
nndep+bidirectional	87s	**0.761**	**0.806**
sr	59s	0.743	0.787
srbeam	102s	0.747	0.794
sr+bidirectional	118s	0.745	0.788
srbeam+bidirectional	175s	0.748	0.794

Table 4.1: Evaluation of Stanford parsing models - MASC corpus (spoken and written)

In our experiments, we compare the performance of Stanford CoreNLP-based DSMs with the performance of DSMs extracted from the same corpora pre-processed with Tree-Tagger (Schmid, 1995) and MALT parser (version `1.8.1`) (Nivre, 2003). We consider this comparison interesting because of the widespread use of Tree-Tagger and MALT parser in NLP tasks and because they ensure good performances at a low processing load and with quick processing speed.[8] MALT parser produces Stanford basic dependencies in a CONLL format, one token per line. In order to keep our experimental design fully factorial, we used the Stanford CoreNLP to convert the basic dependencies produced

[7]The MASC corpus is a manually annotated and balanced subset of 500k words of written texts and transcribed speech sub-section of the Open American Corpus. More detailed information about corpus composition and annotation can be found at `http://www.anc.org/masc`. The comparative evaluation whose results are reported in table 4.1 has been conducted by Thomas Proisl (FAU Erlangen-Nürnberg).

[8]For comparison and to support the interpretation of our experimental results, we evaluated the Tree-Tagger+MALT parser pipeline used for our experiments on the MASC corpus. Even if our results are not really comparable with those in table 4.1 (as the current version of MALT does not support multicore processing, and it produces basic Stanford dependencies), they are in line with predictions, as MALT parser is quicker than Stanford parser and less accurate, but not dramatically so: LAS: 72.8; UAS: 78.6; Processing time: 16 seconds.

by MALT to the CCprocessed format.[9]

For better replicability and since this thesis does not have its primary focus on annotation tools, we did not attempt any task-specific parameter optimization: all tools were used with the most recent off-the-shelf models available with the respective standard distributions.

All sentences longer than 150 tokens were discarded. A quantitative analysis of the distribution of sentence lengths over the three corpora showed that the proportion of sentences longer than 150 tokens was negligible.[10] A closer look at sentences from the Web corpora that are longer than 150 tokens also revealed that they often contain lists of names or web links: from this point of view, the information they contain is not worth the risk of the annotation tools taking very long time or even running out of memory.

4.2 Extraction: from a corpus to a matrix

As discussed in section 2.3.2, the choice of the target vocabulary for the extraction of co-occurrence information is the fundamental preliminary step for the construction of a DSM. We adopted the list of target words from Distributional Memory (Baroni & Lenci, 2010) as the basis of our vocabulary, and we extended it with the experimental items from our experimental datasets described in section 4.5, if not already covered.

Since the extraction of co-occurrence information represents a significant bottleneck in DSM modeling, we extended our vocabulary list with the items from a number of further datasets on which we have not yet conducted experiments, but of general relevance for the DSM and psycholinguistic community. Such datasets are:

- the BLESS dataset (Baroni & Lenci, 2011);
- the SemRel ratings dataset for English (Lapesa, Schulte im Walde, & Evert, 2014);
- the word-pair similarity rating dataset by J. Mitchell & Lapata (2010).

The target vocabulary for all experiments presented in this thesis contains 31132 pos-disambiguated lemmas (20678 nouns, 5080 verbs, 5371 adjectives, 3 adverbs), for a total of 27522 lemmas (without pos-disambiguation).

4.2.1 Context selection function

As discussed in section 2.3.2, we follow Padó & Lapata (2007) in defining contexts as *anchored paths* (a path anchored at the word t starts at t). Surface and dependency paths are extracted from the unparsed and parsed sentence graphs, as a result of the concatenation of the labels of the edges. The **context selection function** operates over the set of the paths anchored at the target t and identifies a subset of paths that are considered potentially informative for the representation of the meaning of t.

In both surface-based and dependency-based models, context selection is based on the following general criteria:

[9] Stanford CoreNLP class `EnglishGrammaticalStructure`.

[10] BNC: 0.03% (average sentence length: 18.6; 75% of the sentences up to 26 tokens); WaCkypedia: 0.05% (average sentence length: 22.6 tokens; 75% of the sentences up to 30 tokens); UkWaC: 0.4% (average sentence length: 25.6 tokens; 75% of the sentences up to 32 tokens).

- Path labels are selected according to the part-of-speech of the connected nodes: only paths linking items from the target vocabulary to **content** words are considered;

- Following Baroni & Lenci (2010) and, to a certain extent, Padó & Lapata (2007),[11] we consider both direct and inverse dependency relations. In the process of extraction of dependency paths, both directions are considered for each edge (e.g, a direct dependency relation between a verb and its subject noun, `nsubj`, and the corresponding inverse dependency relation, `nsubj-1` between the noun and the verb).

- Given two candidate content words a and b in the surface or dependency graph assigned to a sentence, we take into account only the shortest path between a and b; this procedure automatically discards all paths containing cycles.

In this work, only paths connecting content words have been considered. Closed-class words are discarded as potential basis terms, but taken into account in when determining the length of the path in the surface-based model (i.e., they count as intervening word between a target and a potential context word). Similarly, in the dependency-based models, paths are allowed to go through nodes representing words from closed class (e.g., pronouns). Those cases are very limited already, because most of the dependency labels involving function words, as well as punctuation and sentence roots are discarded, as will be discussed in section 4.2.1.2.[12] We do not maintain that these relations bear no semantic content: some of them produce semantic splits that are of clear interest for modeling similarity. Let us consider, for example, negation (`neg`). A negation-sensitive approach in the collection of co-occurrences may lead to improvements in the quality of the co-occurrence vectors (cf., e.g., "Dogs bark" vs. "Dogs don't meow"). However, there are two (interconnected) reasons to discard this type of information at this stage. The first reason is a practical one, it relates to the proliferation of target and context items in the co-occurrence matrix. Since in our approach to the extraction of dependency *paths* we consider function words only if they work as a connector between content words, the only option would be to collect separate co-occurrences for negated and not negated words (e.g., "meow" vs. "NOT_meow"). The result would be very sparse. Second, even if negated and non-negated collocates contribute in a different way to the semantics of their head node, the real linguistic contexts in which negation occurs do not always justify a lexical split between negated and non-negated words (e.g., "Surprisingly, the neighbors' dog did not bark last night"; "Dogs don't bark when they feel safe").

[11] In principle, Padó & Lapata (2007) resort to undirected paths. However, since the path labels used in their study contain ordered information concerning the part of speech of the nodes connected by the edge (e.g., `V:subj:N`, vs. `N:subj:V`) they are specified with respect to their directionality for almost all dependencies (an exception being, for example, nominal modification).

[12] List of discarded dependency labels: `det`, `neg`, `aux`, `auxpass`, `cc`, `cop`, `expl`, `preconj`, `predet`, `punct`, `quantmod`, `root`, `discourse`, `goeswith`, `mark`, `mwe`, `possessive`. For a more detailed description of those dependencies, see the Stanford type dependencies manual (De Marneffe & Manning, 2008). The `prt` relation, which links adverbial particles to the lemma of their head verbs (e.g., "off" to "take" for the verb "take off") was not considered in the construction of dependency paths, but it was exploited to update the lemma of the head verbs. This allows collect co-occurrence information separately for phrasal verbs and their base verbs.

4.2.1.1 Surface-based co-occurrences

As discussed in section 2.2, the notion of window size, commonly employed to characterize the co-occurrence extraction process for bag-of-words DSMs, can be modeled as the length of the path connecting target and basis elements in a precedence graph of the type shown in figure 2.3.

In the surface-based experiments conducted in this thesis, the context selection function takes **path length** within such precedence graph as its parameter, without any further context selection constraint over the label of the edges (LEFT vs. RIGHT). In traditional terms, we adopt a symmetric context window: candidate collocates for the target nodes are searched for in a span of equal length to the left and to the right of the target word. To keep results comparable to those of dependency-based models, we keep co-occurrence extraction within sentence boundaries.

In the surface-based experiments discussed in chapter 6, DSMs are built by incrementally including paths from 1 to 16 edges long. This amounts to say that we consider collocates co-occurring with the target within a maximum span of:

- **one** word to the left and to the right of the target ($\|\pi\| = 1$);
- **two** words to the left and to the right of the target ($\|\pi\| \leq 2$);
- **four** words to the left and to the right of the target ($\|\pi\| \leq 4$);
- **eight** words to the left and to the right of the target ($\|\pi\| \leq 8$);
- **sixteen** words to the left and to the right of the target ($\|\pi\| \leq 16$).

4.2.1.2 Dependency-based co-occurrences

When extracting dependency-based co-occurrences, the selection criteria of the context selection function correspond to the following parameters:

1. **Path length**: only dependency paths up to a certain number of edges selected when constructing the models.
2. **Type of the dependency relations composing the paths**: the syntactic/semantic nature of the dependency relations in each path is exploited as a criterion to build paths deemed to be more closely related to the meaning of the target t (**core** dependencies, main actants of the sentence), or related to it in a mediated way (**external** dependencies, inter-clausal relations and conjuncts).

In the following subsections, we provide some details and motivation for the choice of the values for these two parameters. The end of this section also contains a discussion of the distribution of relation types over path lengths, as well as a comparison of our experimental setup with that of previous work.

Path Length Given a target t, the context selection function operates on the number of dependency edges in each path anchored at the target t For example in the graph displayed in figure 2.4 the dependency path `dobj_amod` anchored the node *barks* and pointing at *cute* (adjectival modifier of its direct object) has length $\|\pi\| = 2$. In the dependency-based experiments discussed in chapter 7, DSMs are built by incrementally including paths from 1 to 5 edges long:

- all paths of length 1 ($\|\pi\| = 1$);
- all paths of length 1 and 2 ($\|\pi\| \leq 2$);
- all paths of length 1, 2, and 3 ($\|\pi\| \leq 3$);
- all paths of length 1, 2, 3, 4, and 5 ($\|\pi\| \leq 5$).

Our choices concerning the path-length parameter are based on the state of the art in the evaluation of syntax-based DSMs, which is here covered and further extended. Earlier work on syntax-based DSMs (G. Grefenstette, 1994; D. Lin, 1998; D. Lin & Pantel, 2001) but also Kiela & Clark (2014) relied on one-step dependencies ($\|\pi\| = 1$). More recent reference studies resort to longer dependency paths: Padó & Lapata (2007) experiment with $\|\pi\| \leq 4$; Baroni & Lenci (2010) do not manipulate path length but employ paths roughly corresponding to $\|\pi\| \leq 3$. For example, the `verb` relation in Baroni & Lenci's (2010) *DepDM*, which labels an underspecified link between the subject of a verb and its complement (e.g., ⟨soldier, `verb`, sergeant⟩ for both "the soldier shot the sergeant" and "the soldier talked with the sergeant"), corresponds to `nsubj-1+dobj` (e.g., ⟨soldier, `nsubj-1+dobj`, sergeant⟩ for "the soldier shot the sergeant") or to `nsubj-1+prep+pobj` (`basic`) and `nsubj-1+prep_*` (`ccprocessed`): e.g., ⟨soldier, `nsubj-1+prep+pobj` or `nsubj-1+prep_to`, sergeant⟩ for "the soldier talked to the sergeant".

Dependency Type According to syntactic-semantic nature of the relations encoded, we classified the dependencies between content words into two groups.

The first group contains dependencies which link to direct arguments or adjuncts of the main predicate of the sentence, as well as their non-clausal modifiers. The encoded relations are fully syntagmatic, and they establish connections among the main actants of the sentence. In this work, dependencies from the first group are labelled as `core`:[13]

- Subjects: active (`nsubj`) and passive (`nsubjpass`); only in the CCprocessed dependencies: agents (`agent`), i.e., complements of passive verbs introduced by the preposition *by*;
- Objects: direct (`dobj`), indirect (`iobj`), and prepositional (`pobj`);
- Adjectival complements of verbs (`acomp`);
- Predicative complements of verbs and adjectives (`pred`);
- Prepositions with noun phrase complements (`prep`); only in the CCprocessed dependencies: lexicalized prepositions and multiword prepositions with noun phrase complements (e.g., `prep_on`, `prep_with`, `prep_from_behind`);
- Modifiers: adjectival (`amod`), adverbial (`advmod`), temporal (`tmod`); noun phrases functioning as modifiers (`npadvmod`); nominal compound modifiers (`nn`);
- Numerals (`num`) and numeric modifiers (`number`);

[13] For more details on the specific relations, as well as for examples of use, see the Stanford Typed Dependencies manual (De Marneffe & Manning, 2008).

- Possessors (`poss`).

A second group of dependencies interclausal relations and conjunctions/appositions. The encoded relations are not directly at the linguistic `core` of the sentence, and they have both a syntagmatic and paradigmatic nature. In this work, dependencies from the second group are labelled as `extra`:

- Clausal subjects: active (`csubj`) and passive (`csubjpass`);
- Clausal complements of a verb or an adjective with their own subjects (`ccomp`) and without their own subjects (`xcomp`); only in the CCprocessed dependencies: controlling subjects of clausal complements without subjects (`xsubj`);
- Relative clause modifiers of a noun phrase (`rcmod`);
- Relations between the main verb of a relative clause and the head of a Wh-phrase (`rel`); this relation occurs only with relative words which are neither the subject nor the object of the relative word, as those cases are analysed as `nsubj` and `dobj`, respectively;
- Reduced non-finite verbal modifiers that are neither core arguments of a verb nor full finite relative clauses (`vmod`);
- Prepositions with clausal complements (`pcomp`); only in the CCprocessed dependencies: lexicalized prepositions with clausal complements (e.g., `prepc_without`);
- Adverbial clause modifiers of a verb phrase or sentence (`advcl`);
- Conjunction relations between two elements (`conj`); only in the CCprocessed dependencies: lexicalized conjunction relations between two elements (e.g., `conj_and`, `conj_or`);
- Appositional modifiers, parenthesized examples, and defining abbreviations (`appos`);
- Parataxis relations between the main verbs of a clause and other sentential elements without any explicit coordination or subordination (`parataxis`).

In our experiments, paths are labelled as `core` if they contain only `core` dependencies, and as `extra` if they contain at least one `extra` dependency. In our experiments, we compare the performance of dependency-based DSMs built from `core` dependencies to that of DSMs built from core and extra dependencies (`core+extra`, abbreviated as `extra`).

The (dependency-based) context selection function in practice Let us take the sentence graph in figure 4.5 as an example to illustrate the process of path extraction and selection. Table 4.2 lists the paths anchored at the target *immigration*, selected and classified according to the context selection criteria described in this section (*Length* stands for path length, *Type* stands for dependency type).

As described at the beginning of this section, we always select the shortest paths between two nodes. In the example sentence, two paths are available between the target *immigration* and the node *Bills* (lemmatized as *bill*): the one-step path ⟨immigration,

Length	Path	Type
1	⟨immigration, `prep_on-1`, bill⟩	`core`
	⟨immigration, `conj_and-1`, port⟩	`core`
2	⟨immigration, `prep_on-1+nsubjpass-1`, submit⟩	`core`
3	⟨immigration, `prep_on-1+nsubjpass-1+agent`, Brownback⟩	`core`
4	⟨immigration, `prep_on-1+nsubjpass-1+agent+nn`, senator⟩	`core`
	⟨immigration, `prep_on-1+nsubjpass-1+agent+appos`, republican⟩	`extra`
5	⟨immigration, `prep_on-1+nsubjpass-1+agent+appos+prep_of`, Kansas⟩	`extra`

Table 4.2: Context selection, dependency graph from figure 4.5: paths anchored at the node *immigration*

`prep_on-1`, bill⟩ and the two-step path ⟨immigration, `conj_and-1+prep_on-1`, bill⟩. Only the shortest of the two paths was selected, namely ⟨immigration, `prep_on-1`, bill⟩.

Comparison between the dependency graph in figure 4.5 and the selected paths shows also that the `auxpass` link was excluded from the computation of paths. Only two paths belong to the `extra` dependency type (for the presence of the `appos` relation).

Tables 4.3, 4.4, and 4.5 display the distribution of dependency groups (columns) over path lengths (rows) for each corpus, parser, and dependency type. The data for paths of length 1 correspond to the overall distribution of dependency types in our models. For example, table 4.3 reports that for the BNC parsed with the MALT parser in the basic format, the core dependencies represent the 68.19% of the dependencies, while they are proportionally more significant for the same path length in the CCprocessed format (76.63%). This trend, constant across corpora and parsers, is not surprising because dependency collapsing introduces direct and lexicalized links (e.g., `prep_in`) between heads and prepositional complements which correspond to two-step paths in the basic representation, and propagation of conjuncts introduces new edges to the original graph. We also report the number of dependencies (column *Dep*) that the parser identified but was unable to classify (`dep` relation label). Although experiments on underspecified paths could be interesting (as those paths contain lower quality information which can be considered as "noise" in the parsed data), we had to keep our parameter space manageable and therefore discarded them.

For paths longer than one-step, the *Core* columns report the percentage of paths of length n that are composed only of core dependencies, while the *Extra* columns contain the percentage of paths that are composed of both core and extra dependencies, and the *Dep* columns report the percentage of paths containing an underspecified relation. The proportion data show how the ratio between core and extra paths is heavily affected by path length: a not surprising effect given that paths containing `extra` dependencies necessarily build on subpaths containing the `core` ones.

One final observation can be made, which concerns the distribution of `extra` dependencies over corpora. While sentences in WaCkypedia and UkWaC are longer than sentences in BNC, BNC still contains proportionally more `extra` dependencies than the other two corpora.

Len	MALT parser						Stanford parser					
	Basic			CCprocessed			Basic			CCprocessed		
	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep
1	68.19	27.38	4.41	76.63	19.58	3.78	68.07	28.05	3.86	75.39	21.12	3.48
2	46.85	43.13	10.00	51.22	39.45	9.31	47.51	44.62	7.85	51.38	40.04	8.57
3	31.70	53.85	14.43	29.41	55.18	15.40	32.52	56.15	11.31	30.49	55.54	13.95
4	21.76	59.67	18.55	16.05	62.49	21.44	22.46	62.93	14.59	17.42	63.53	19.03
5	14.93	62.57	22.48	7.42	64.58	27.98	15.46	66.80	17.73	8.95	66.68	24.36

Table 4.3: BNC: distribution of path groups over path lengths (MALT vs. Stanford parser; Basic vs. CCprocessed dependencies)

Len	MALT parser						Stanford parser					
	Basic			CCprocessed			Basic			CCprocessed		
	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep
1	72.50	23.29	4.19	80.51	16.09	3.39	72.29	23.97	3.73	79.19	17.46	3.33
2	51.48	39.11	9.39	56.19	35.30	8.49	51.63	40.53	7.82	57.09	34.35	8.54
3	35.61	50.25	14.13	33.14	51.99	14.86	36.05	52.07	11.86	35.43	49.73	14.83
4	25.30	56.23	18.45	18.43	60.06	21.50	25.64	58.63	15.72	20.74	58.03	21.22
5	17.65	59.53	22.81	8.23	62.64	29.11	17.88	62.64	19.47	10.24	61.54	28.21

Table 4.4: WaCkypedia: distribution of path groups over path lengths (MALT vs. Stanford parser; Basic vs. CCprocessed dependencies)

Len	MALT parser						Stanford parser					
	Basic			CCprocessed			Basic			CCprocessed		
	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep	Core	Extra	Dep
1	68.28	25.79	5.91	76.42	18.60	4.96	68.00	26.54	5.45	74.82	20.23	4.93
2	47.28	39.29	13.41	51.88	35.65	12.45	47.58	40.50	11.91	52.16	35.29	12.53
3	28.70	50.25	21.04	27.73	50.21	22.05	29.16	51.87	18.96	29.08	48.74	22.17
4	19.72	53.59	26.68	15.16	54.37	30.45	20.02	55.62	24.34	16.49	52.89	30.61
5	12.93	54.80	32.25	.77	53.98	39.24	13.11	57.23	29.64	8.04	52.63	39.32

Table 4.5: UkWaC: distribution of path groups over path lengths (MALT vs. Stanford parser; Basic vs. CCprocessed dependencies)

4.2.2 Path value, basis mapping, and co-occurrence quantification

In the experiments reported in this thesis, the *path value* function is not parametrized: both in window-based and dependency-based DSMs, all paths anchored at the target t contribute to the same extent to its distributional profile.

As far as *basis mapping* is concerned, the corresponding function maps targets into *lemmatized* version of the basis terms, in both surface-based and syntax-based DSMs. For both classes of models, we compare *word-based mapping* to *structured mapping*. Within window-based models, the use of an undirected context window is an instance

of *word-based mapping*, with target words being mapped to basis context words (e.g., CUTE, BONE, EAT for the target *dog* in the sentence *the cute dog ate the bone*). Directed context windows represent an instance of *structured mapping*, because target words are mapped into context dimensions which contain a direction label defining the portion of the context relative to the target in which the collocate was found (e.g., LEFT+CUTE, RIGHT+EAT, RIGHT+BONE for the target *dog*). Within syntax-based models, we compare dependency-filtered models (*word-based mapping*) to dependency-structured models (*structured mapping*). In the first case, context dimensions correspond to words (e.g., BONE and EAT for the target *dog* if the context selection function only allows subjects and objects), and path properties are only exploited as criteria for the context-selection function. In the second case, the path label is concatenated to the word at which the path ends, producing more linguistically informed context dimensions (e.g., SUBJ-1+OBJ+BONE and SUBJ-1+EAT for the target *dog*).

4.3 Manipulation of the co-occurrence matrix

In this section, we discuss the parameters related to the manipulation of the co-occurrence matrix to improve the quality of the vector representations, and motivate our choices concerning the parameter values. We refer the reader to section 2.3.3 for a detailed description of the these parameters.

Score for feature weighting We compare plain co-occurrence *frequency* to *tf.idf* and to the following association measures: *Dice coefficient*; *simple log-likelihood*; *Mutual Information*; *t-score*; *z-score*. We selected these measures because they have widely been used in previous work on DSMs (tf.idf, MI and log-likelihood) or are popular choices for the identification of multiword expressions. Based on statistical hypothesis tests, log-likelihood, t-score and z-score measure the significance of association between a target and feature term; MI shows how much more frequently they co-occur than expected by chance; and Dice captures the mutual predictability of target and feature term. Note that we compute sparse versions of the association measures with negative values clamped to zero in order to preserve the sparseness of the co-occurrence matrix. For example, our MI measure corresponds to Positive MI in the other evaluation studies. See Evert (2008) for a thorough description of the association measures and details on their calculation (Fig. 58.4 on p. 1225 and Fig. 58.9 on p. 1235). When calculating association measures for dependency-typed models, marginal frequencies for feature terms are calculated at the level of basis terms. In practice, this means that in the experiments involving structured basis mapping, namely directed window for bag-of-words and dependency typed for syntax-based, marginal frequency for the context is calculated for each feature label (i.e., RIGHT+dog, SUBJ+dog); refer to section 2.3.3 for a discussion of alternative approaches to the calculation of expected frequencies in dependency-structured models.

Feature transformation To reduce the skewness of feature scores, it is possible to apply a transformation function. We evaluate *square root*, *sigmoid* (tanh) and *logarithmic* transformation vs. *no transformation*. Note that transformation functions are applied additionally to frequency or association measures, independently of the application of a transformation in the calculation of the association score (i.e., to MI, which

is the result of a logarithmic transformation of the ratio between observed and expected co-occurrence frequency).

4.3.1 Dimensionality reduction

In this section, we define the parameter space related to the dimensionality reduction strategies evaluated in our experiments. As discussed in section 2.3.4, dimensionality reduction can be performed by means of feature selection (i.e., retaining only a subset of the matrix columns) or feature extraction (i.e., low-rank matrix factorization with PCA, SVD or NMF), or by applying feature extraction on a subset of the features of the original matrix.

Dimensionality reduction: feature selection In our experiments, we perform feature selection based on marginal frequency. For both window-based and dependency-based experiments, we rank the matrix columns according to the marginal frequency of the basis terms, and we select the top *5k, 10k, 20k, 50k, 100k* dimensions. Note that, while this step is usually performed before feature scoring/transformation for efficiency reasons, it belongs, conceptually, to the dimensionality reduction step in the pipeline. The experiments on window-based DSMs involved an additional feature selection parameter, namely the **feature selection criterion**: we compared *marginal frequency* to *number of nonzero co-occurrence counts* as a ranking criterion for feature selection. The number of non-zero co-occurrence counts for a basis term b corresponds to the number of unique targets with b as a feature.[14] As it will be shown in chapter 6, we found no significant difference between the two parameter values (ranking of dimensions according to frequency vs. number of non-zero counts). To keep the parameter space manageable, we decided to exclude this parameter from the evaluation of dependency-based DSMs.

Dimensionality reduction: feature extraction Section 2.3.3.2 and 2.4 reviewed the dimensionality reduction strategies commonly adopted in DSM. We focus on SVD because among the available methods, it is the only one which produces ordered dimensions.[15] This has clear advantages: from a practical point of view, it allows to experiment with different reduced dimensionalities without rerunning the reduction algorithm; from a theoretical point of view, the rank of the SVD dimensions is a criterion for the experimental manipulation of the SVD parameters; indeed, the effect of selecting dimensions based on their rank (e.g., keeping vs. discarding the first dimensions) has already been explored in the literature (Bullinaria & Levy, 2012), opening interesting research directions that the experiments presented in this aims at pursuing by framing SVD into a larger-scope parameter set.

[14] Feature selection according to non-zero counts can be applied before or after the calculation of (sparse) association measures. The general need to keep the parameter space manageable requires limiting the number of experiments. In this particular case, calculating the number of non-zero counts based on marginal frequency is the basic option: it is the natural counterpart of feature selection based on frequency, which is what is commonly done in distributional semantics, while we are not aware of studies in which matrix columns are selected based on their global mass of association scores.

[15] A further advantage of SVD with respect to RI and BEAGLE is the connection between the co-occurrence information in the input and the reduced representation. This adds the results in Lapesa & Evert (2013a) as a further reason to prefer SVD to RI: the latter was found to clearly underperform the former (albeit in a more restricted experimental setting than the one presented in this thesis). As for BEAGLE, it just does not scale to larger corpora (Sahlgren et al., 2008).

With the output of feature selection described in the previous section as a starting point, we optionally apply Singular Value Decomposition to 1000 dimensions, implemented with randomized SVD (Halko et al., 2011) for performance reasons, as some of the involved matrices are very large.[16]

In our experiments, we evaluate two parameters which regulate the use of the dimensionality reduced matrix for the computation of similarity.

- **Number of latent dimensions**: out of the 1000 SVD dimensions, we select the first *100*, *300*, *500*, *700*, *900* dimensions (i.e. those with the largest singular values);
- **Number of skipped dimensions**: when selecting the reduced dimensions, we discard the first 50 or 100 dimensions and we compare the performance of achieved by discarding the first dimensions to that of the full reduced matrix (parameter value: 0). This parameter has already been evaluated by Bullinaria & Levy (2012), who achieved best performance by discarding the initial components of the reduced matrix, i.e., those with the highest variance.

4.4 Projecting meaning in space

Distance Measures In the experiments reported in this thesis, the range of evaluated metrics is restricted to *cosine distance* (i.e., angle between vectors) and *Manhattan distance* (Minkowski metric with $p = 1$, also known as L1 distance). We adopt such restricted range for a number of reasons:

- Both conceptually and practically in the DSM evaluation pipeline, the computation of similarity/distance comes as the last step; in the need of keeping the parameter space manageable, we decided to adopt a restricted set of distance metrics, and conduct a more thorough evaluation of the previous steps; further work can then build on robust tendencies for best/worse parameters identified up to this point;
- Cosine is considered a standard choice in DSM modeling and is adopted by most evaluation studies (Bullinaria & Levy, 2007, 2012; Polajnar & Clark, 2014);
- For our normalized vectors, Euclidean distance is fully equivalent to cosine;

[16] Randomized SVD (rSVD) exploits randomization methods to make SVD computation more efficient, especially for large matrices. Given a matrix A with shape $m \times n$, to be reduced to a desired dimensionality k, rSVD decomposition is achieved as follows. A subspace of intermediate dimensionality, B, is produced, which captures most of the action in A, i.e., the vectors in B live in the image space of the vectors in A. In practice, this is achieved by producing a random matrix O with shape $m \times (k \times o)$, which is multiplied with A producing an orthogonal projection in a space of intermediate dimensionality; the oversampling factor, o, regulates the number of dimensions of the intermediate subspace B: as soon as the oversampling factor enlarges, the resulting reduced matrix approximates the full SVD. The algorithm allows to further update B to further pulling the vectors towards the image space of A; this operation can be repeated multiple times (number of power iterations). SVD is performed on B, which is now dense and low-dimensional $(k \times o)$.

The training parameters for rSVD are k, o, and n. In the experiments presented in this thesis k is set to 1000. Following the recommendation of Halko et al. (2011) and the default in the `wordspace` package, and after some preliminary experiments which confirmed the stability of the results, both o and n were set to 2.

- Preliminary experiments with the Maximum distance measure (Minkowski metric with $p \to \infty$, also known as Canberra distance) resulted in very low performance.

Relatedness in the semantic space Given two words a and b represented in a DSM, we consider two alternative ways of quantifying the degree of relatedness between a and b. The first option (and standard in DSM modeling) is to compute the *distance* (cosine or Manhattan, as described in the previous paragraph) between the vectors of a and b. The alternative choice, evaluated in our experiments is based on *neighbor rank*. As already discussed in section 2.3.4.2, neighbor rank has not yet been evaluated systematically neither with respect to the tasks, nor with respect to its potential interaction with other parameters. For the multiple choice tasks, we compute *rank* as the position of the target word among the nearest neighbors of each word holding the potential relation of interest (e.g., for TOEFL, we calculate the rank of target among the neighbors of the candidate synonym). Note that using the positions of the candidate word among the neighbors of the target would have been equivalent to direct use of the distance measure, since the transformation from distance to rank is monotonic in this case. For the correlation and clustering tasks, we compute a symmetric *rank* measure as the average of $\log \text{rank}(a, b)$ and $\log \text{rank}(b, a)$. An exploration of the effects of directionality on the prediction of similarity ratings and its use in clustering tasks (i.e., experiments involving $\text{rank}(a, b)$ and $\text{rank}(b, a)$ as indexes of relatedness) is left for future work.

4.5 Selection of word similarity tasks

Chapter 3 already provided an overview of the tasks employed for the evaluation of DSMs in general. In this section, we now describe specifically the evaluation tasks and datasets that are used in the experiments presented in this thesis. In particular, the experiments of this thesis have been conducted on the following three types of evaluation tasks.

Multiple choice classification task Distributional relatedness between a target word and two or more other words is used to select the best, that is, the most similar, candidate. Performance in this task is quantified in terms of decision **accuracy**. Evaluation is conducted on a number of datasets that encode different types of semantic relations. The first dataset, from the well-known **TOEFL** multiple-choice synonym test (80 items; Landauer & Dumais, 1997), encodes synonymy. It is also included in most of the reference DSM evaluation studies (e.g., Bullinaria & Levy, 2007; Padó & Lapata, 2007; Baroni & Lenci, 2010; Bullinaria & Levy, 2012; Kiela & Clark, 2014). The other datasets are the semantic priming datasets introduced in section 3.4: the **GEK** dataset (404 items), which encodes event-based relatedness and has already been employed in Lapesa & Evert (2013a,b,c), and the **Semantic Priming Project** datasets (963 items in total), which encode different types of relations (synonymy, antonymy, co-hyponymy, backward and forward phrasal association) and are evaluated in this thesis for the first time.

Correlation task Distributional relatedness between the representations of target words is compared to native speaker judgments of semantic similarity or relatedness.

Following previous studies (Baroni & Lenci, 2010; Padó & Lapata, 2007), performance in this task is quantified in terms of **Pearson** correlation.[17] Evaluated datasets are the **Rubenstein and Goodenough dataset** (RG65) of 65 noun pairs (Rubenstein & Goodenough, 1965), also evaluated by Padó & Lapata (2007); Baroni & Lenci (2010); Kiela & Clark (2014), and the **WordSim-353 dataset** (WS353) of 353 noun pairs (Finkelstein et al., 2002), included in the study of Polajnar & Clark (2014).

Clustering task We employ distributional relatedness to assign words to a predefined set of semantic classes and quantify DSM performance in terms of **purity**. We evaluated the same clustering datasets as Baroni & Lenci (2010): the **Almuhareb-Poesio set**, the **Battig set** (Van Overschelde et al., 2004), the **ESSLLI 2008 set**, and the **Mitchell set** (T. Mitchell et al., 2008). Clustering is performed with an algorithm based on partitioning around medoids (Kaufman & Rousseeuw, 1990, ch. 2), using the R function `pam` with standard settings. Other clustering studies have often been carried out using the CLUTO toolkit (Karypis, 2003) with standard settings, which corresponds to spectral clustering of the distributional vectors. Unlike `pam`, which operates on a pre-computed dissimilarity matrix, CLUTO cannot be used to test different distance measures or neighbor rank. Comparative clustering experiments showed no substantial differences for cosine similarity; in the rank-based setting, `pam` consistently outperformed CLUTO clustering. See Appendix A for more details.

4.6 Computational tools

In this section, we list the computational tools employed to carry out the experiments reported in this thesis. The brief outline provided here follows the steps for the extraction/evaluation of DSMs which has been established in chapter 2 and followed again in sections 4.1-4.4.

- Pre-processing: see section 4.1 for a detailed description of the pre-processing tools used in the experiments.
- Extraction of co-occurrence information: surface-based co-occurrences for window-based DSMs have been extracted with the IMS Corpus WorkBench[18] and the UCS toolkit;[19] dependency-based co-occurrences have been extracted from the parsed corpora with the `networkx` package[20] for Python.
- Manipulation of co-occurrence matrices, similarity computation, as well as evaluation experiments have been implemented in R using the `wordspace` package

[17] The presentation of the evaluation results focusses primarily on Pearson correlation; some other evaluation studies, as discussed out in chapter 3, adopt Spearman's rank correlation ρ, which is more appropriate if there is a non-linear relation between distributional relatedness and human judgements. We computed both coefficients in our experiments and decided to report Pearson's r for two reasons: (i) Baroni & Lenci (2010) already list r scores for a wide range of DSMs in this task, and their evaluation was the reference by the time the research project presented in this thesis has started; (ii) linear regression analyses for ρ and r showed comparable trends and patterns for all DSM parameters.

[18] `http://cwb.sourceforge.net/index.php`

[19] `http://www.collocations.de/software.html`

[20] `https://networkx.github.io/`.

(Evert, 2014). The experiments have been run on the High Performance Computing cluster at the University of Erlangen-Nürnberg[21] and on the servers of the IMS Stuttgart.

4.7 Summing up

This chapter discussed the experimental setup of the evaluation experiments presented in this thesis. For each dataset, we tested all parameter combinations. This resulted in:

- Window-based DSMs: 537600 model runs (generated and evaluated within approximately 5 weeks on a high performance cluster);
- For syntax-based DSMs: 806400 model runs for each dependency-filtered and dependency structured (generated and evaluated within approximately 6 weeks on a high performance cluster).

[21] https://www.rrze.fau.de/serverdienste/hpc/

5

Interpreting DSM performance

This chapter addresses a very important issue often disregarded in the DSM literature: the methodology used for the interpretation of the results of the evaluation experiments. Note that evaluation methodologies are completely independent of the experimental setups, which have been the topic of the previous chapters. In fact, it is often possible to apply different interpretation strategies to the same set of experimental results.

As has been discussed in Chapter 4, the magnitude of the parameter set tested in this thesis together with the full factorial design adopted in the evaluation (involving all combinations of parameter values) resulted in a very large amount of experimental runs. Such a large number of experiments makes a robust statistical methodology for the interpretation of the results particularly necessary. A typical way of conducting DSM evaluation is to look at the best DSM configurations: in other words, the criterion for model selection is to pick the experimental run with the highest performance. This strategy is clearly not applicable in the context of this thesis, as it is at high risk of overtraining, given the large number of experiments: a certain parameter configuration may be the best fit for a certain dataset, but this may be due to chance and not to specific properties of the selected parameters. An additional drawback of this procedure is that it is likely to fail in the identification of robust trends characterizing the interactions between DSM parameters. The evaluation methodology proposed in this thesis successfully overcomes both of these issues.

We employ linear regression as a statistical tool to understand the impact of different parameters on model performance. DSM parameters and their interactions are considered predictors of model performance. In this way, we achieve a solid understanding of the impact of specific parameters and parameter interactions on DSM performance, which can inform the selection of DSM settings that are robust to overfitting, as it is possible that the best run in terms of absolute performance will turn out to be overtrained and therefore disregarded because it is not generalizable enough.

This chapter is structured as follows. Section 5.1 reviews the evaluation approaches adopted in the DSM literature, focussing on large-scale reference studies, compares them in terms of their impact on the scope of the evaluation, interpretability of the results, and points out the main points of strength and weakness. Section 5.2 introduces the linear regression methodology employed in this thesis and spells out the key statistical concepts which will be employed throughout the thesis. Section 5.3 complements the theoretical introduction with a toy example, illustrating the `R` implementation and providing the guidelines for understanding the evaluation plots in the subsequent chapters.

5.1 Standard approaches

In this section, we review different approaches adopted in the literature for the interpretation of results of DSM evaluation experiments. A systematic review of various evaluation studies reveals the following three major approaches.

One model many tasks This approach is used for testing a single new model with fixed parameters or a small number of new models. The evaluated model is tested on a range of tasks and is compared to competing models, applying little or no parameter tuning.

Examples of this approach are the studies of Padó & Lapata (2007) (Dependency Vectors) and Baroni & Lenci (2010) (Distributional Memory). Both studies have been conducted to assess the potential of dependency structured DSMs, at a point in time when little previous work on the topic existed. Padó & Lapata (2007) employ the Rubenstein and Goodenough dataset as a development set to tune a number of parameters (e.g., feature weighting, distance metric, path-value function) and pick the best performing model as a fixed setting for their main experiments, which involve a number of evaluation tasks as well as a comparison to the state of the art. Baroni & Lenci (2010) test different degrees of lexicalization in the construction of dependency-structured DSMs and, additionally, they introduce a novel strategy for the quantification of co-occurrences (type-based co-occurrence counting defined in section 2.3.2.4). In total, they evaluate three versions of Distributional Memory on a large selection of tasks and compare them to the state of the art.

Incremental tuning An alternative approach to evaluation is adopted in studies which aim at testing a large number of DSM parameters, but typically do not introduce new ways of constructing a DSM. While the studies described in the previous paragraph have as their main goal that of showing that a certain novel DSM performs better than its state-of-the-art competitors, the studies discussed in this section address the question of which parameter combination ensures the best DSM performance. The need for the identification of a best model brings in a crucial methodological issue: model selection. The set of evaluated parameters and their values defines a search space which needs to be explored to find the best combination of parameter values. A possible way of exploring this search space is by means of **incremental tuning**: parameters are tested sequentially to identify their best performing values. Incremental tuning proceeds in steps of one parameter (i.e., once parameter a is tuned, proceed to tune parameter b) or in pairs of parameters (e.g., once parameter a is tuned, proceed to set b and c). Optionally, the best setup across tasks is identified by averaging DSM performance over all tasks.

The incremental tuning approach characterizes the studies by Bullinaria & Levy (2007, 2012); Polajnar & Clark (2014); Kiela & Clark (2014). Bullinaria & Levy (2007) report a systematic study of the impact of a number of parameters (shape and size of the co-occurrence window, distance metric, association score for co-occurrence counts) on a number of tasks (including the TOEFL synonym task, which is also evaluated in this thesis). Their evaluated models are built from the British National Corpus. Bullinaria & Levy (2012) extend the evaluation reported in Bullinaria & Levy (2007): starting from the optimal configuration identified in their first study, they switch to ukWaC as a

source corpus and test the impact of three further parameters (application of stop-word lists, stemming, and dimensionality reduction using Singular Value Decomposition) on a number of tasks (including TOEFL and clustering on the dataset from T. Mitchell et al. (2008), also evaluated in this thesis). Polajnar & Clark (2014) evaluate the impact of context selection (for each target, only the most relevant context words are selected and the remaining vector entries are set to zero) and vector normalization (used to vary model sparsity and the range of values of the DSM vectors) in standard tasks related to word and phrase similarity. Kiela & Clark (2014) evaluate window-based and dependency-based DSMs on a variety of tasks related to word and phrase similarity; a wide range of parameters are involved in this study: source corpus, window size, number of context dimensions, use of stemming, lemmatization and stop-words, similarity metric, score for feature weighting.

The main drawback of this approach is that it is not fair to all involved parameters, as there is no guarantee that a parameter value which is discarded at earlier stages of the incremental model selection would not turn out to be stronger in combination with other parameters which ought to be tested only later. Moreover, even when incremental evaluation proceeds in pairs of parameters, it can only reveal a limited amount of parameter interactions. In other words, an evaluation conducted by means of incremental tuning is heavily dependent on the order in which parameters are tested, which is established by the experimenter and which brings in a number of assumptions and potential biasing factors.

Testing all parameter combinations An alternative to the incremental tuning approach is to explore all parameter combinations with a **full factorial design** and to pick the best setup per task. Given the same set of parameters and values, the incremental tuning approach described in the previous paragraph requires less runs than the full factorial one: e.g., for two parameters a and b with 2 and 4 values respectively, incremental tuning requires 6 model runs (2 to set a, plus 4 to set b based on the best value of a), while testing all combinations requires 8 runs. This often has practical consequences on the scope of the evaluation: incremental tuning studies can afford to explore a larger parameter space than the full factorial ones. This should not be considered an advantage though, because incremental tuning explores the parameter space only partially, as discussed in the previous paragraph.

The full factorial approach is adopted in recent evaluation studies targeting the comparison between count and predict DSMs, namely in the studies by Baroni, Dinu, & Kruszewski (2014) and Levy et al. (2015). Baroni, Dinu, & Kruszewski (2014) experiment with 36 count DSMs (manipulating size of the context window, score for feature weighting, dimensionality reduction) and 48 cbow predict DSMs (manipulating size of the context window, number of reduced dimensions, and other `word2vec`-specific parameters). The comparison also involves count-vectors from Distributional Memory, predict vectors from Collobert et al. (2011), as well as state of the art vectors for the respective tasks. The evaluation covers standard semantic similarity datasets, which are also evaluated in this thesis, and analogy datasets. Levy et al. (2015) compare PPMI-based count models (72) to embeddings generated with SVD (432), skipgram `word2vec` (144), and GloVe (24). In total, 672 models are evaluated on the prediction of word similarity ratings and analogies. Evaluated parameters regulate context window, computation of association metrics, and post-processing of the embeddings (e.g., eigenvalue weighting,

normalization). Best setups are compared to the results achieved in a vanilla scenario (all parameters set to default) and with the recommended configuration of `word2vec`. From a methodological perspective, the evaluation by Levy et al. (2015) can be considered the most interesting among the existing ones, and that for two reasons. The first reason is the scope of the evaluation: the authors define a core of parameters which can be aligned across methods due to theoretical and practical considerations (e.g., number of negative samples in skipgram and the shifting parameter in a PPMI DSM; context distribution smoothing in skipgram and smoothed PPMI). As a result, 72 parameter configurations are shared across methods (if we exclude GloVe, whose parameter space is quite limited anyhow). The second reason is the model selection strategy, as the authors compare the best evaluation run per task to the result of parameter tuning by two-fold cross-validation (parameters are tuned on a half of each dataset, tested on the other half, and final evaluation scores are calculated by averaging the two runs for each data point). Their results show that, often, two-fold cross-validation correctly identifies the optimal configuration, in particular for larger datasets.

While potentially fair to all evaluated parameters, the studies adopting a full factorial design and targeting the best setup per task lack in interpretability and face the risk of overfitting, as discussed in the introduction of this chapter. As far as parameter interactions are concerned, the full factorial design in principle allows to capture them, but the fact that the above studies focus on the best parameter combinations leaves this issue unaddressed. Finally, identifying the best setup across tasks by averaging the performance of different DSMs has the drawback that the interpretation may be highly biased by the (possibly overtrained) performance on a specific task.

5.2 Interpreting performance with linear regression

In the previous sections we identified the two requirements that a suitable DSM evaluation methodology needs to meet: be robust to overtraining and be capable of capturing interactions between parameters. The evaluation methodology proposed in this thesis successfully addresses both issues.

As already pointed out earlier, our study adopts a full factorial design: we tested all parameter combinations for the window-based and dependency-based DSMs, respectively. Differently from the studies listed in section 5.1, however, we do not look for the best parameter combinations but employ linear regression to interpret the impact of different parameters on DSM performance. The goal of this section is to define the theoretical building blocks of the proposed methodology; refer to Harrell (2015) for a more detailed introduction to regression modeling techniques and an overview of their application to different data analysis cases.

We use linear regression to analyze the influence of individual parameters on DSM performance using general linear models with performance as a dependent variable (Y) and model parameters as independent variables (p_1, p_2):

$$\mathrm{Y} = \beta_0 + \beta_1 \cdot p_1 + \beta_2 \cdot p_2 + ... + \beta_n \cdot p_n + \epsilon$$

The weights (coefficients, β_1, β_2) learnt by the linear model represent the impact of the predictors on the predicted variable: a positive impact will be encoded in positive weights (predicted performance is higher than mean performance, encoded in the intercept β_0) or negative ones (predicted value lower than β_0). ϵ is an error term, which

represents the discrepancy between the actual value and the value predicted by the linear model; model training seeks to find the combination of weights which minimizes ϵ.

Turning to a more concrete example, the following equation quantifies the effect of the manipulation of source corpus and window size on the performance in the TOEFL multiple-choice task:

$$\text{accuracy} = \beta_0 + \beta_{corpus} + \beta_{window} + \epsilon$$

When categorical predictors are involved in the regression, the linear model learns a weight for each distinct value of the predictor (in statistical terms, for each level of a factor). For each predictor, one value is considered as reference, and the weights of the others are calculated in relation to it. In our example, β_{corpus} is a set of three weights: with BNC as a reference value, β_{bnc}=0; β_{wacky} and β_{ukwac} quantify the expected gain/loss in predicted performance with respect to models built from the BNC. Because of the dummy coding β_0 corresponds to the DSM accuracy when all predictors are set at their reference level, and it does not correspond to the grand mean of the accuracy (as in the case of continuous predictors).

We code all predictors as categorical: parameters with numerical values (e.g., size of the context window, number of dimensions) are considered as discrete factors. This choice is motivated by the need of minimizing the number of assumptions, as we expect the relation between performance and predictors to be neither linear nor monotonic: for example, we have no reason to assume the difference in performance at window 2 and 16 to be eight times bigger than the difference between 2 and 4. Indeed, our analysis show that the shape of the effects is often complex (e.g., performances reaches a peak for an intermediate value of a parameter and degrades for larger values). To account for such complex non-linear shapes we would need to fit polynomial models with a large number of parameters (potentially, as many parameters as the values of the predictors), increasing the risk to overfit the data.

Specific combinations of parameter values can affect DSM performance to a different degree: in our example, WaCkypedia could be the best corpus overall, but perform poorly with the smallest context window. In order to identify combinations of parameter values which positively affect DSM performance, we introduce in our regression model all two-way interactions among parameters:

$$\text{accuracy} = \beta_0 + \beta_{corpus} + \beta_{window} + \beta_{corpus:window} + \epsilon$$

There is no theoretical reason to restrict the analysis to two-way interactions; in fact, it is mainly for practical reasons that in the analyses presented in the subsequent chapters we focus on interactions between pairs of parameters, and leave higher-order interactions for future work.

5.3 In practice

The application of linear regression to the evaluation data allows us to address two main questions: a) What are the parameters that affect DSM performance the most? b) What are the best parameter values, i.e., those with robust effects across model runs?

In this section, we illustrate our evaluation methodology by spelling out the steps of the analysis on a small subset of the evaluation data which will be analyzed in the subsequent chapters. While the purpose of this section is mostly didactic, we also provide the details for the implementation of the analysis in `R`, as well as the criteria for the interpretation of tables and plots that we will use in the subsequent chapters.

Dataset As a case study for this toy example, we selected the evaluation on the TOEFL task, on window-based models, without dimensionality reduction. We will conduct the analysis on a subset of the full parameter space; we selected 75 runs resulting from the manipulation of 3 parameters, namely source corpus, size of the context window, and number of context dimensions. The other parameters have been set to default values.[1] Accuracy ranges from 42.50 to 70.00, with a mean of 57.28.[2] In what follows, we assume that the subset is stored in a table (a dataframe, in R): we refer to it as `toefl`.

Fitting and evaluating the linear model In this example, we test the explanatory power of source corpus and window size in our dataset, and train two linear models: one without interactions (`m1`) and one with interactions (`m2`):

```
m1 <- lm(accuracy ~ corpus + window, data=toefl)
m2 <- lm(accuracy ~ corpus + window + corpus:window, data=toefl)
```

A preliminary question is whether our linear models are doing a good job at predicting DSM performance: only in case of a positive answer we can build reliable generalizations from them. A commonly employed diagnostic for linear model fit is R^2. It compares the actual values of the dependent variable with the prediction of the model and provides an estimation of whether the model has learnt a good approximation of the data on which it has been trained. R^2 is calculated as the ratio between the variance captured by the regression and the total amount of variance in the data, and therefore ranges between 0 and 1 (perfect fit).[3]

It has been pointed out that R^2 tends to overestimate model fit when the sample size is small and a large number of parameters are involved (more parameters always explain more variance). It is therefore recommended to rely on the adjusted version of R^2, which takes into account sample size and number of predictors in the model. Adjusted R^2 is always smaller than R^2, can have negative values and does not have a fixed range.[4] The difference between R^2 and Adjusted R^2 tends to be large for small sample sizes and neglectable for large sample sizes.[5]

[1] Default values: window direction: undirected; criterion for the selection of context dimensions: frequency; score for feature weighting: frequency; transformation: log; distance metric: cosine; index of distributional relatedness: distance.

[2] The distribution of accuracy in this subset of the data is not representative of the one of the full dataset (where, for example, maximum accuracy is 87.5).

[3] Unadjusted R^2 is also referred to as *squared correlation* because (under a number of assumptions which are all met in our case) it is equal to the square of the correlation between the actual and predicted values.

[4] Adjusted R^2 is calculated as follows: $Adj.R^2 = R^2 - (1-R^2)\cdot(p/(N-p-1))$, with N equal to sample size and p as the number of parameters of the model. For a thorough discussion of the mathematical motivation behind the calculation of Adjusted R^2 see Harrell (2015).

[5] In our case, sample size corresponds to the number of different parameter combinations (e.g.,

In R the fit of a linear model can be explored with the `summary()` function. We display a selection of its output (omissions are marked as `[...]`). We do not report nor discuss all details, for which we refer the reader to an introduction to statistical analysis with R, e.g., Baayen (2008).

```
summary(m1)

[...]
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  64.6333     0.8511  75.938  < 2e-16 ***
corpuswacky   1.9500     0.7880   2.475   0.0158 *
corpusukwac   1.9000     0.7880   2.411   0.0186 *
window2      -1.8333     1.0173  -1.802   0.0760 .
window4      -8.0833     1.0173  -7.946 2.75e-11 ***
window8     -15.1667     1.0173 -14.909  < 2e-16 ***
window16    -18.0833     1.0173 -17.776  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
[...]
Multiple R-squared:  0.8797,        Adjusted R-squared:  0.869
[...]
```

Of clear relevance for the purpose of this section are the R^2 (`Multiple R-squared`) and `Adjusted` R^2 values, as well as the estimates for intercept (β_0 in the equations in section 5.2) and all levels of the predictors (e.g, `corpuswacky` and `corpusukwac` correspond to β_{wacky} and β_{ukwac} in section 5.2, respectively).

As discussed before, the coefficients are employed to construct the predictions of the model. For example, to calculate the predicted accuracy for corpus = ukwac and window = 4, we add up `(Intercept)`, `corpusukwac` and `window4`: 64.6333+1.9000+(-8.0833) = 58.45.

Table 5.1 displays a subset of the `toefl` dataframe, updated with the predictions of the model both without interactions (column *pred_m1*) and with interactions (column *pred_m2*). For each datapoint we also show the squared deviation of the accuracy from the overall mean (column *s_data*), and the squared deviation of the predicted accuracy from the actual accuracy (columns *s_m1* and *s_m2* for `m1` and `m2`, respectively). The *sum* row reports the sum of the squared deviations in *s_data*, *s_m1*, and *s_m2*, respectively), for the entire dataset.

The ratio between the sum of the squared deviations of the predicted values (`s_m1` and `s_m2`) and the variance in the actual data (`s_data`) quantifies the amount of variance that the model did not manage to account for, corresponding to the R^2 of the ϵ term in the regression equation. (Unadjusted) R^2 of the regression model can therefore be calculated as $1 - R^2_\epsilon$. Let us spell out the calculations for our toy dataset:

$$\texttt{m1}\text{: } R^2 = 1 - (527.79/4385.5) = 0.8797$$
$$\texttt{m2}\text{: } R^2 = 1 - (229.37/4385.5) = 0.9477$$

N=33600 in an unreduced setting). A linear model trained on the all TOEFL experimental runs in an unreduced setting, with all parameters as predictors as well as all two-way interactions achieves an R^2 of 0.874 and an Adjusted R^2 of 0.873.

corpus	window	cont.dim	accuracy	pred_m1	pred_m2	s_data	s_m1	s_m2
ukwac	4	5000	61.25	58.45	57.75	15.73	7.84	12.25
ukwac	4	10000	58.75	58.45	57.75	2.15	0.09	1.00
ukwac	4	20000	57.50	58.45	57.75	0.05	0.90	0.06
ukwac	4	50000	55.00	58.45	57.75	5.21	11.90	7.56
ukwac	4	100000	56.25	58.45	57.75	1.07	4.84	2.25
bnc	16	5000	42.50	46.55	44.25	218.55	16.40	3.06
bnc	16	10000	42.50	46.55	44.25	218.55	16.40	3.06
bnc	16	20000	43.75	46.55	44.25	183.15	7.84	0.25
bnc	16	50000	46.25	46.55	44.25	121.73	0.09	4.00
bnc	16	100000	46.25	46.55	44.25	121.73	0.09	4.00
...	...	...	...	...	...	...	...	...
SUM						**4385.5**	**527.79**	**229.37**

Table 5.1: Toy example: actual vs. predicted accuracy

As expected, `m2` has a better fit to the data than `m1`. Adjusted R^2 values confirm that the models are not overfitting our dataset (without interactions: 0.869; with interactions: 0.935).

Feature ablation To quantify the importance of each parameter or interaction we adopt a feature ablation setting: for each parameter, we measure its impact on DSM performance in terms of the amount of explained variance for which it is directly responsible (both as a simple effect, and because of the interactions in which it participates).

The feature ablation value for a specific parameter represents the drop in R^2 that we would observe if we were to remove the parameter from the regression model, keeping in mind that dropping a parameter means to drop also all the interactions in which the parameter is involved (e.g., dropping the parameter a automatically discards the interaction $a : b$). Feature ablation is calculated for interactions, as well: if the model only contains two-way interactions, the feature ablation value of an interaction corresponds to its R^2; in case of high order interactions the feature ablation value of an interaction includes the R^2 of its higher-order terms (e.g., dropping the two-way interaction $a : b$ automatically drops the three-way interaction $a : b : c$).

We implement feature ablation by conducting analysis of variance, which is straightforward for our factorial design as it decomposes the R^2 of the full model into the partial R^2 for each term. We employ the `anova()` function from the `rms` package. `anova.rms()` is a convenient choice because the sum of squares for each predictor is already calculated jointly for the predictor and all the interactions in which it participates. In what follows, we focus on the linear model with interactions and show a subset of the output of `anova.rms()`:

```
library(rms)
# Fit linear models with ols(), as required by rms()
ol_2 <- ols(accuracy ~ corpus + window+corpus:window, data=toefl)
# Run anova
anova(ol_2)
```

```
                Analysis of Variance          Response: accuracy

 Factor                                              d.f. Partial SS [...]
 corpus  (Factor+Higher Order Factors)               10    360.2083
  All Interactions                                    8    298.4167
 window  (Factor+Higher Order Factors)               12   4094.3750
  All Interactions                                    8    298.4167
 corpus * window  (Factor+Higher Order Factors)      8    298.4167
 REGRESSION                                          14   4156.1667
 ERROR                                               60    229.3750
```

The `Partial SS` values show that window accounts for a very large amount of the variance explained by the regression model (`REGRESSION`) (cf. the calculations in table 5.1); corpus, on the other hand, accounts for a small (yet highly significant, according to the values produced by the anova function and not shown here for reasons of space) amount of the variance. Furthermore, a very large portion of variance explained by *corpus* is actually due to its interaction with *window* (298.4167 out of 360.2083 Partial SS, corresponding to 0.07 and 0.08 R^2, respectively).

The `plot(anova.rms())` function automatically transforms Partial SS values into R^2 and it can be used to produce feature ablation plots which show parameters on the y-axis and partial R^2 on the x-axis.[6] The plot in figure 5.1 is the default for `plot(anova.rms())`: it will be employed throughout the thesis to display feature ablation for main effects, with minor cosmetic changes.[7]

```
plot(anova(ol_2), what="partial R2", sort=c("ascending"), margin=c())
```

Partial R^2 can also be displayed in a tabular format, as shown in the code below. Note that the values do not sum up to the R2 of the full model, because `corpus * window` contributes to the partial R^2 of both `window` and `corpus`.

```
df_ols <- as.data.frame(plot(anova(ol_2), what="partial R2"))
names(df_ols) <- "partial R2"
df_ols
```

```
                partial R2
window          0.93360759
corpus          0.08213543
corpus * window 0.06804557
```

[6] A note of caution is necessary with feature ablation experiments based on `plot(anova.rms())` on small datasets, as the implementation is based on R^2 instead of Adjusted R^2 and therefore potentially biased towards parameters with a larger number of factors. However, our sample sizes are very large and, as discussed before, there is a very small difference between R^2 and Adj. R^2. We conducted a sanity check on the full TOEFL dataset and compared ablation scores based on R^2 and Adj. R^2, for the parameters with the highest number of values: window size, number of context dimensions, and score. As we expected, the difference between the two values is always well below 0.01: window, $R^2 = 0.2144$; Adj. $R^2 = 0.2152$; score, $R^2 = 0.3945$; Adj. $R^2 = 0.3957$; number of context dimensions, $R^2 = 0.0178$; Adj. $R^2 = 0.0176$.

[7] The method also displays a number of other diagnostics for the contribution of each predictor (e.g.. chi-square, p-value, etc.): they have been suppressed here with the command `margin=c()`.

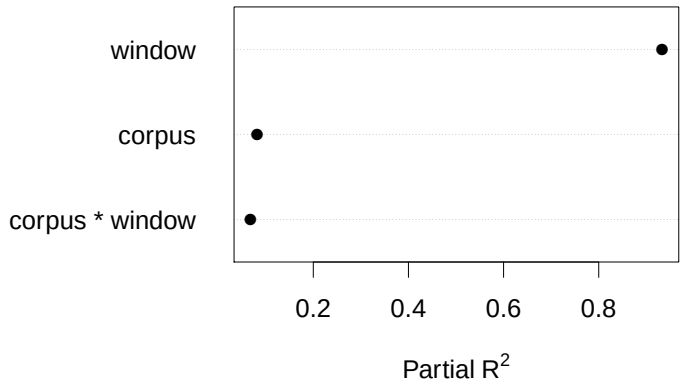

Figure 5.1: Toy example: feature ablation plot

In the subsequent chapters, we will employ the dot plots to visualize feature ablation of specific parameters, and will display the most powerful interaction per task in a tabular format.

Before moving on to the choice of best parameter values, let us clarify the nature of our feature ablation experiments. Feature ablation is commonly employed in Data Analysis for stepwise model selection: starting point is a large set of candidate predictors for the dependent variable, and the goal is to identify a subset of predictors which does the best job at modeling the dependent variable. Feature ablation is a widely employed technique in Machine Learning, as well: in this case, starting point are a set of features and a task, and the goal is to identify the features which ensure the best performance on the task. Our approach is different, in that our main interest does not target best parameters (best predictors, best features), but best parameter values; moreover, given our full factorial design, a stepwise selection would not be motivated, as the dropping of a predictor would not change the picture of the relative performance of the other parameters. Feature ablation informs our choice of best parameters: if a parameter turns out to have little or no effect on DSM performance (low feature ablation), it can be set to a default value, usually the one which is less computationally expensive (a smaller corpus, for example); if the parameter has a big impact on model performance (high feature ablation value), we proceed to explore its best parameter values as discussed in the following section. It should be kept in mind that a comparable feature ablation value for a certain DSM parameter in two different tasks (e.g., `window` in TOEFL vs. WS353) only indicates that the parameter affects DSM performance *to the same extent*, but it does not necessarily imply that the same parameter values need to be selected to ensure the best performance.

Finding best parameter values To identify the best DSM parameter values, we exploit the predictive power of linear regression and manipulate the values of the DSM parameters to quantify the expected gain/loss in performance associated to their values. Under the assumption that a linear model with a good fit to the data can be considered a sort of "smoothing" algorithm which is capable of extracting robust trends

by filtering away random noise, we build our interpretation based on the predictions of the model (e.g., predicted accuracy).

We identify robust optimal parameter settings with the help of effect displays (Fox, 2003) as implemented in the `effects` library. Effect plots are particularly convenient especially for models involving interactions because, unlike coefficient estimates, they allow an intuitive interpretation of the effect sizes of categorical variables irrespective of the dummy coding scheme used.

Effect displays show the partial effect of one or two parameters by marginalizing over all other parameters (e.g., by taking the average of the predictions for parameters that are not shown). When displaying interactions (`corpus:window`), the main effects marginal to each interaction (`corpus` and `window`) are incorporated when producing the effect plot; this allows the predictions of the interaction to range over the values of the main effects, providing a more accurate estimation of the overall effect of the manipulation of the two parameters.

The code below produces plots displaying the effect of corpus and window on predicted accuracy (displayed on y-axis). First, we produce the the simple effect plots for corpus (figure 5.2) and window (figure 5.3). Then we plot their combined effect for the model without the interaction (figure 5.4) and for the one with interactions (figure 5.5); parameter values are displayed on the x-axis for the simple effects, and in combination with the different line styles for the model containing the interaction.

```
library(effects)
# Corpus (simple effect)
plot(Effect(c("corpus"), m1), ci.style="none")
# Window (simple effect)
plot(Effect(c("window"), m1), ci.style="none")
```

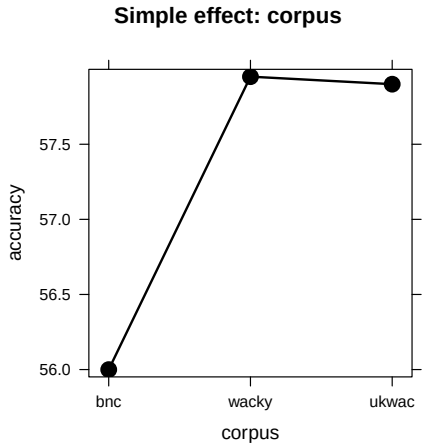

Figure 5.2: Simple effect: corpus

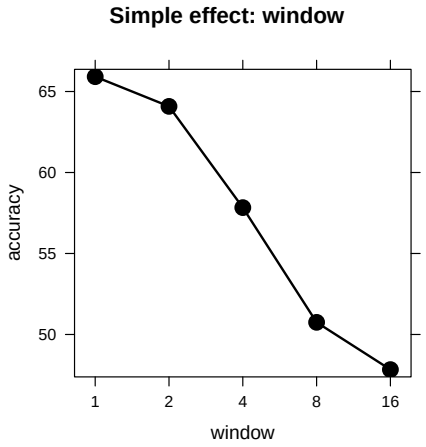

Figure 5.3: Simple effect: window

```
# No interactions: corpus and windows as simple effects
plot(Effect(c("corpus", "window"), m1), multiline=TRUE)
# With interactions: joint effect of corpus and windows
plot(Effect(c("corpus", "window"), m2), multiline=TRUE)
```

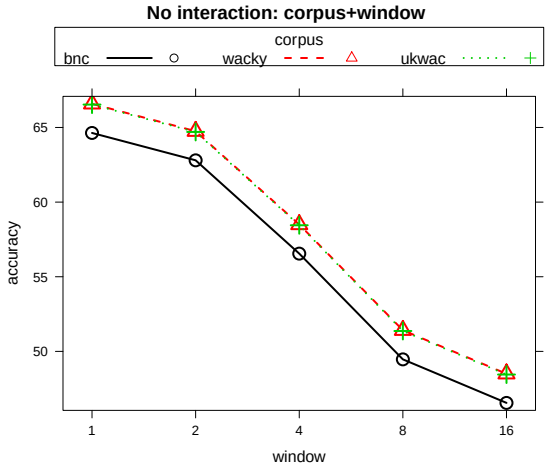

Figure 5.4: Combination of simple effects

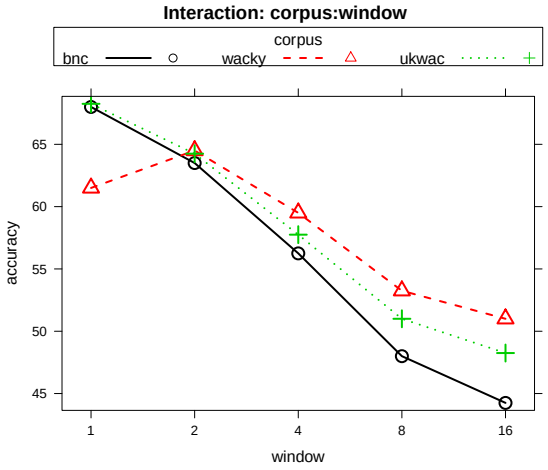

Figure 5.5: Interaction

The comparison between figure 5.2 and 5.3 illustrates the different explanatory power of the two predictors: the range of the displayed predicted accuracy in 5.2 is very restricted, compared to the one of 5.3. This means that the manipulation of `corpus` does not result in variation in the performance of a DSM. The comparison between figure 5.4 and 5.5 aptly illustrates the contribution of `corpus:window` to the model fit. Manipulation of window size affects predicted performance differently for each corpus; the interaction is particularly salient for the WaCkypedia corpus: the performance pattern is quite different from the one captured by the model without interactions (figure 5.4). If we were to choose best parameter values based on the effect plot in figure 5.4, we would go for WaCkypedia as a source corpus, and a context window of size 1. The interaction plot in 5.5, however, reveals that window size 1 would be an unfortunate choice for WaCkypedia, and that the misleading main effect is due to the averaging behind the construction of the main effects.

Note that, given that *corpus* and *window* are the only predictors in our models and simple effects are absorbed into interactions, figure 5.4 and 5.5 fully represent `m1` and `m2`, respectively. For this reason, plotted values correspond to the predicted values generated with the `predict()` method, as shown above.

The range of the predicted accuracy varies across the plots, as an outcome of the different explanatory power of the different predictors: compare the partial R^2 values for corpus and window (0.93 vs. 0.08) with the range of the y-axis in the respective effect plots (2 vs. 17 points). When interpreting effect plots, we recommend the reader to focus on the differences between parameter values, and not to their predicted accuracy.

The effect plots shown in the subsequent chapters of this thesis will differ from default settings of the `effects` library only for cosmetic details, and can therefore be interpreted according to the guidelines discussed above.

5.4 Summing up

In this chapter, we illustrated a novel methodology for the interpretation of DSM performance. It is based on linear regression with performance as dependent variable and

DSM parameters as independent variables. We argued that, differently from other evaluation approaches adopted in the DSM literature, our methodology can capture interactions between parameters and serve as a basis for robust generalizations concerning best parameter settings, avoiding overfitting. The theoretical discussion of the statistical properties of our methodology has been complemented with a toy example on a small dataset, based on real evaluation data. The aim of such toy example is twofold: first, it can be used as a reference for the interpretation of the plots employed in the subsequent chapters of this thesis; second, as all crucial steps are spelled out and accompanied by the corresponding `R` code, it constitutes a documentation for users who want to apply the analysis to their own evaluation data.

6

Evaluation of window-based DSMs: Word similarity tasks

In this chapter we discuss the results of the evaluation of window-based DSMs on word similarity tasks. Part of the material presented here has been published in Lapesa & Evert (2014a): at the level of the scope of the evaluation, this chapter integrates the publication with the evaluation of the DSMs without dimensionality reduction; at the level of interpretation, we provide here more extensive discussions and comparisons.

Given the large number of runs, maximum or mean performances are not very informative: our discussion will therefore not involve the distribution of performances, but we refer the reader to the histograms displayed in appendix C.

Unsurprisingly, our results show that, besides the practical advantages in memory usage and computation speed, SVD improves DSM performance (or, at least, does not have a detrimental effect). There are, however, several reasons that make an in-depth analysis of unreduced runs interesting. First, from the practical point of view, there are applications which require non-negative (possibly interpretable) dimensions (e.g., measures of distributional inclusion; information-theoretic measures) which will benefit from a better understanding of the window-based parameters. Second, from a theoretical point of view, we show that a comparison of the impact of different parameters in an unreduced vs. reduced setting reveals interesting properties of the SVD reduction, and that it should not be taken for granted that the best unreduced space is also the best input for SVD. Last but not least, the vectors of an unreduced DSM are a collection of co-occurrences, and the interaction between specific parameters can inform the discussion on the properties of association measures (e.g., robustness to low-frequency effect), independently on their application to DSM purposes: from this point of view, DSM modeling can be seen as an extrinsic evaluation of underlying corpus linguistic methods.

In the following sections we guide the reader through the interpretation of the evaluation results. The discussion is structured per evaluation setting: we start from the multiple-choice task (TOEFL, section 6.1), proceed to discuss the prediction of similarity ratings jointly for two datasets (RG65 and WS353, section 6.2), and conclude with the clustering experiments (AP, BATTIG, ESSLLI, MITCHELL, section 6.3). In each section, we compare reduced and unreduced runs. We first discuss feature ablation results and then proceed to select best parameter values by interpreting the effect plots for the relevant parameters. Given that a systematic evaluation of the index of distri-

butional relatedness is one of the main contributions of this thesis, we explore its effect in a dedicated section, where we compare this effect across the different datasets, for reduced and unreduced runs (section 6.4). While discussing best parameter values, we will always be reasoning in terms of higher or lower *predicted performance*: as discussed in chapter 5, we assume that if the fit of the model is good, predicted performances can to be considered as a good generalization of the actual ones. In section 6.5 we bridge the gap between predicted and actual performance, and check the *actual* performance of the best settings which we identify based on the effect plots. A summary of the main findings of this study, in form of recommended settings for the different tasks, is provided in section 6.6.

6.1 TOEFL

The distribution of performance of our DSMs in the TOEFL task is displayed in the histograms in figure 6.1 (unreduced runs) and 6.2 (reduced runs): more specifically, the histograms visualize on the y-axis the number of models (i.e., distinct parameter settings) which reached a the accuracy values displayed on the x-axis. The histogram also report the minimum, maximum, mean performance, as well as standard deviation.

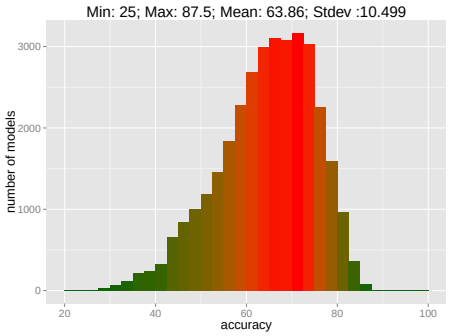

Figure 6.1: TOEFL, unreduced runs

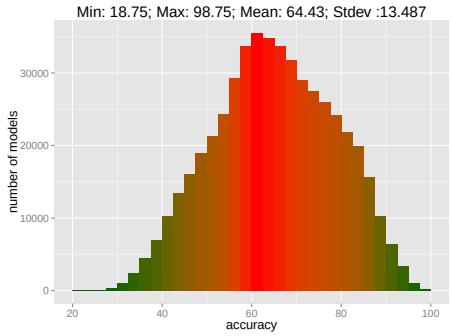

Figure 6.2: TOEFL, reduced runs

The histograms show that, in both cases, we observe a large number of relatively good models, in particular in the SVD-reduced experiments: this is, however, not very informative. The distributions display a high variability and, given the large number of runs, we are not really interested in which DSMs we find at its extremes. However, we are interested in the factors that determine the performance of our DSMs *across the whole distribution*: as discussed in chapter 5, we believe that our linear regression methodology will allow us to identify *robust trends across our parameter space*, avoiding to pick *just one model* just because it happens to be the best one. For this reason, from now on in the thesis we will not display distribution of performance anymore: the interested reader will find all the histograms in appendix C.

Let us now turn to the feature ablation analysis of the relative impact of the DSM parameters, which we already outlined in chapter 5. Tested in the task of predicting DSM accuracy on the TOEFL dataset, the linear models achieve an adjusted R^2 of 87% (unreduced runs) and 89% (reduced runs), respectively. The plots in figure 6.3 display the ranking of the evaluated parameters according to their importance in the feature ablation setting, for the unreduced (left) and reduced (right) runs. Parameters with a high feature ablation value are those whose parameter values make a difference

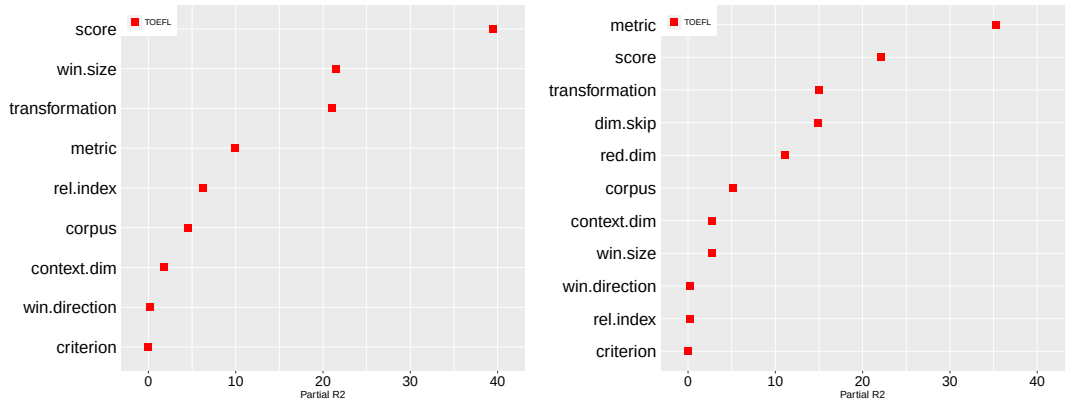

Figure 6.3: TOEFL: Feature ablation. Left: unreduced; Right: reduced.

Interaction	df	R^2
score:transformation	18	7.38
transformation:metric	3	1.70
win.size:transformation	12	1.31
win.size:score	24	1.01
score:metric	6	0.96
corpus:score	12	0.88
score:context.dim	24	0.86
corpus:win.size	8	0.61

Interaction	df	R^2
score:transformation	18	7.42
metric:dim.skip	2	4.44
score:metric	6	1.77
metric:context.dim	4	0.98
win.size:transformation	12	0.91
corpus:score	12	0.84
score:context.dim	24	0.64
metric:red.dim	4	0.63

Table 6.1: TOEFL: interactions, R^2. Left: unreduced; Right: reduced.

in the predicted performance: in other words, those for which it is particularly crucial to pick the right value (or combination of values, in case of interactions). As already discussed in chapter 5, the R^2 values in the plots refer to the proportion of variance explained by the respective parameter together with all its interactions, corresponding to the reduction in R^2 if this parameter is left out. The feature ablation values of all interactions above 0.5 R^2 is displayed in the tables in 6.1.

On the basis of their influence in determining model performance, we identify three parameters that are crucial for the TOEFL task, both in a reduced and in an unreduced setting: *feature score*, *feature transformation*, and *distance metric*. The tables in 6.1 show that these parameters are involved in a high number of interactions, among which the strongest is the one between score and transformation. Throughout the thesis, we will show that these parameters affect the distributional space independently of tasks and datasets. Their interactions are better understood by looking at the effect plots: we will elaborate more on it while discussing best parameter values, and we will show that it is possible to identify a set of score/transformation combinations with robust performance across all tasks.

The two SVD-related parameters are also powerful, in particular the *number of skipped dimensions*: this is not unexpected and confirms the findings by Bullinaria & Levy (2012), who achieved state-of-the-art in TOEFL discarding the first SVD dimensions.

Two parameters exhibit a drop in their feature ablation value (relative to other parameters), from an unreduced to an reduced setting: *window size* and, more dramatically, *relatedness index* (which drops to $R^2 < 0.5$). SVD makes the manipulation of

these parameters less influential: for *window size*, we will show that the drop is due to the fact that detrimental values (i.e., large windows) are rescued by the SVD projection; in the case of *relatedness index*, the fact that SVD reduces the difference between distance and rank is to be interpreted as a reduction in the overall asymmetry of the space: as this is a tendency which affects all tasks, we will discuss it more extensively in section 6.5. Interestingly, *relatedness index* is not involved in any strong interaction, showing that its contribution is constant across the different values of other parameters.

The impact of *corpus* and *number of context dimensions* is intermediate in both settings, but it is ranked slightly higher in the SVD setting. As our corpora differ in size, we can take the difference in the relative ranking as an indication of the fact that SVD is sensitive to the amount of input data: when discussing best parameter values, we will come back to this point, as well.

Exclusion criterion and, to a lesser extent, *direction of the context window* have a weak explanatory power ($R^2 < 0.5$), showing that their manipulation does not heavily affect DSM performance.

6.1.1 Best parameter values

Unreduced setting While the main goal of this section is to identify best parameter values for TOEFL in the unreduced setting, it is also meant to serve as an illustration of the interpretation of DSM performance based on effect plots. We talk the reader through all the interactions listed in table 6.1, allowing some redundancy to give a full picture, and discuss extensively the criteria for the selection of the best parameter values. Starting from the following sections, we will present just a selection of partial effect plots and refer the reader to the supplementary material for a full picture. Keep in mind that effect plots display the relative effect of a parameter on the *predicted performance*: what we will be discussing from now on, throughout the entire thesis, will be differences in predicted performance (according to the weights learnt by our linear models), calculated by manipulating the parameter (or the pair of parameters) of interest, and averaging across the parameters that are not shown.

Let us start with the interaction between *feature score* and *feature transformation*, displayed in figure 6.6. It is particularly instructive to compare the interaction plot with the plots of the corresponding simple effects (figure 6.4 and 6.5). These plots show how main effects result from averaging across the entire parameter space but fail to capture the fact that best values vary a lot in combination with other parameter values. In our example, while *no transformation* is much better than *log* and *root* as a main effect, it is overperformed by *log* and *root* when it comes to *simple-ll*. If we were to set the best values based on main effects, we would go for *simple-ll* and *no transformation*. The interaction shows us that the best choice is *z-score* with *no transformation*, and if *simple-ll* needs to be selected (for example for a task-unspecific best setting) then it should combine with *log* or *root*.

The best results are achieved by association measures based on significance tests (*simple-ll, t-score, z-score*), followed by *MI*. This result is in line with previous studies (Bullinaria & Levy, 2012; Kiela & Clark, 2014), which found PMI or PPMI to be the best feature scores, and consistent with the predictions from the literature on the statistical properties of association measures:

- Association measures identify the best collocates for the target words, and there-

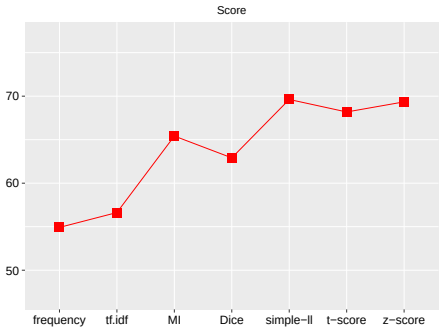

Figure 6.4: TOEFL, unred: score

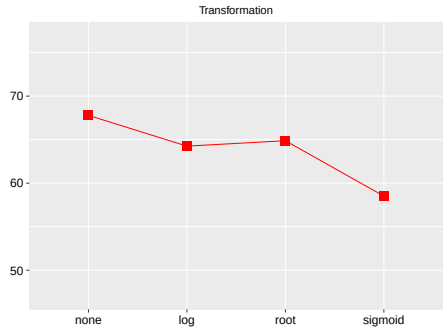

Figure 6.5: TOEFL, unred: transf

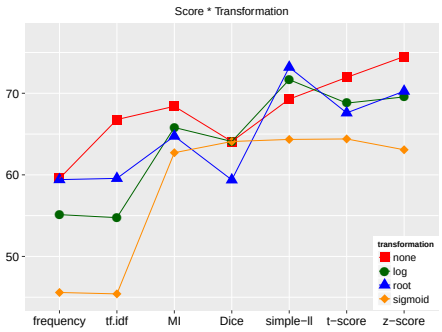

Figure 6.6: TOEFL, unred: score / transformation

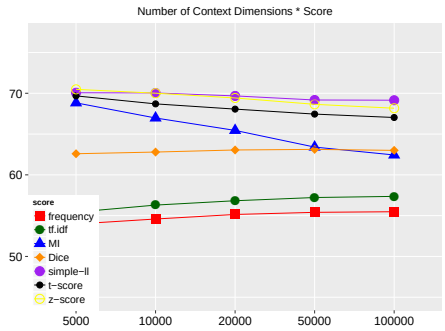

Figure 6.7: TOEFL, unred: score / context.dim

fore they do a better job than frequency at highlighting their most salient features. It is no wonder that this improves the semantic representation encoded in a DSM.

- The association measures involved in our experiments quantify both attraction and repulsion between target and the features, resulting in negative scores that are then cut of at zero. This increases the sparseness of the resulting space: Polajnar & Clark (2014) labelled the increase of the sparseness of the DSM vector as "thinning" (refer to the discussion in section 2.3.3.1 for the relation between thinning and positivization). They do not differ in terms of the degree of introduced sparseness. Differently from the measures discussed so far, *Dice* and *tf.idf* cannot increase the sparseness of the of the distributional space.

- Association measures are known to differ in terms of their robustness to low-frequency data. In particular, *MI* and, to a lesser extent, *z-score* are known to overestimate node/collocate pairs containing low-frequency items, while *simple-ll* and *t-score* are more robust to low frequency effects. This tendency is partially confirmed in the figure 6.7, which displays the interaction between *feature score* and *number of context dimensions*. A smaller number of context dimensions corresponds to a more aggressive filtering of low-frequency features: *MI* is the measure for which this effect is more marked, *simple-ll* the one which is more robust to its manipulation.

- Of the involved association measures, *simple-ll* is the one which produces the most skewed scores (extremely high values compared to other association measures). In our experiments on TOEFL, it is the only measure for which a root or log

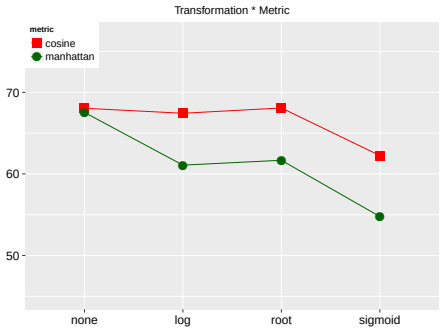

Figure 6.8: TOEFL, unred: transformation / metric

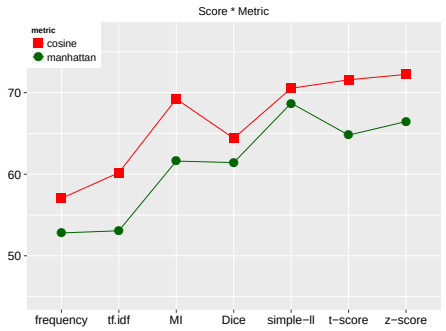

Figure 6.9: TOEFL, unred: score / metric

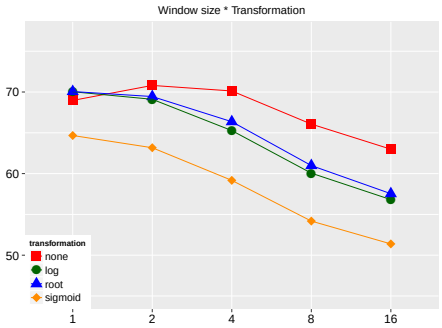

Figure 6.10: TOEFL, unred: win.size / transformation

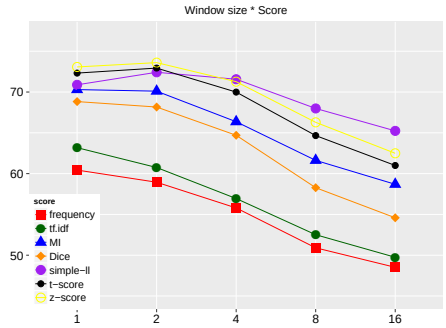

Figure 6.11: TOEFL, unred: win.size / score

transformation is recommended, while other the other measures do not require a particular transformation.

- Of all involved transformations, *sigmoid* is the one which displays a consistently detrimental effect. The poor performance may be due to the fact that a sigmoid transformation is, in fact, a soft binarization which translates the high variability of the feature scores into a scale from 0 to 1, resulting in information loss. The only score for which *sigmoid* is not detrimental is *Dice*, which is already on a 0 to 1 scale.

Summing up, the partial effects plots inspected so far suggest that best performances are achieved with *z-score*, *no transformation*, and *5k* or *10k* context dimensions. A valid alternative is *simple-ll*, with *root* transformation, with *5k* or *10k* context dimensions.

Let us now turn to the *distance metric*, whose effect is displayed in interaction with *feature transformation* in figure 6.8 and *feature score* in figure 6.9. The interaction with *feature transformation* suggests that, while *cosine* is the most robust choice across transformations, *manhattan* is equivalent to it when no vector transformation is involved; the interaction with *feature score*, however, shows that *cosine* is the recommended choice in all cases. The best *distance metric* is therefore *cosine distance*: this is one of the consistent findings of the evaluation presented in this thesis and it is in accordance with Bullinaria & Levy (2007) and, to a lesser extent, Kiela & Clark (2014).[1]

[1]In Kiela & Clark (2014), *cosine* is reported to be the best similarity metric, together with the

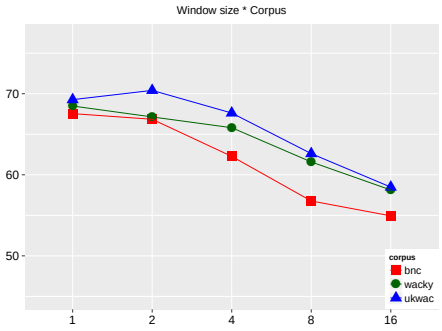

Figure 6.12: TOEFL, unred: corpus / win.size

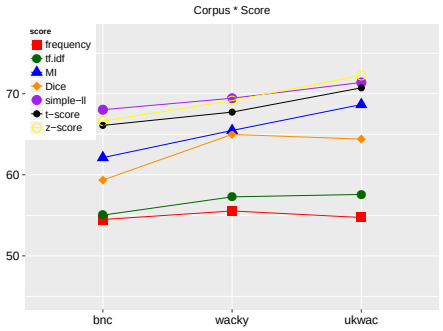

Figure 6.13: TOEFL, unred: corpus / score

Next, we explore the *window size* parameter, based on its interactions with *feature transformation* (figure 6.10) and *feature score* (figure 6.11). Choosing a small window (size 1 or 2) improves DSM performance, larger windows have a detrimental effect: the overlap of the most immediate context is relevant for the modeling of synonymy, while context features occurring farther in the sentence only introduce noise. Joint inspection of the interaction plots in figures 6.10 and 6.11 identifies in a window of size 2, without any transformation, the best option. Among the association measures, *z-score* which we already identified as the best choice in the interaction with transformation, predicts its best performances at a window size of 2. *Simple-ll* exhibits a higher degree of robustness to the manipulation of the context window: the detrimental effect of a window of size 4 is only minimal, as compared to the other involved measures. This is an interesting property to keep in mind in the perspective of the individuation of a best setting *across tasks*, and it illustrates one of the main strengths of the methodology proposed in this thesis, as the inspection of partial effect plots makes it possible to find reliably robust "second best" choices.

Figures 6.12 and 6.13 display the interaction between *source corpus* and *window size* and *feature score*, respectively: the larger corpus, *UkWaC*, is clearly the best choice across the possible combinations.

Association measures based on statistical tests (*simple-ll*, *z-score*, *t-score*) and *MI* appear to be the right choice to exploit the co-occurrence information coming from larger corpora, with increasingly good predicted performances as compared to *frequency* or *tf.idf* which are robust to corpus choice, but perform comparably poorly.

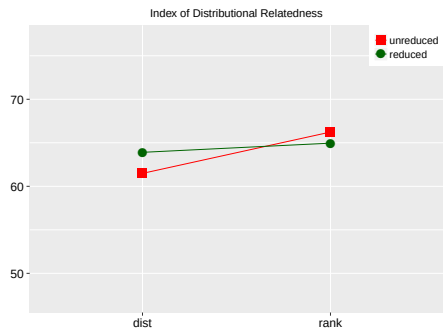

Figure 6.14: TOEFL, rel.index

We now turn to *index of distributional relatedness*: the absence of strong interactions with other parameters allows to reliably inspect the main effect (figure 6.14). It shows that *neighbor rank* is the best choice in the unreduced setting and anticipates

correlation similarity metric (a mean-adjusted version of cosine similarity). The latter, however, turned out to be more robust across different corpora and weighting schemes.

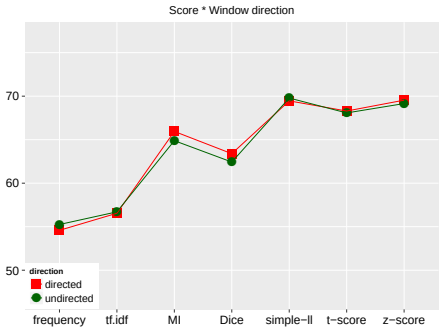

Figure 6.15: TOEFL, unred: direction / score

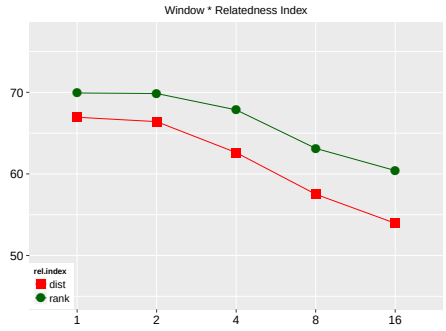

Figure 6.16: TOEFL, unred: window / relindex

that the same holds for the reduced setting, too, albeit to a weaker extent (recall the low R^2 value, below our threshold of 0.5). We refer the reader to section 6.4 for a more extended discussion of the dynamics of the effects involving relatedness index, as well as for comparisons across tasks.

Let us conclude by showing two additional interactions which did not enter the interaction tables because of their low R^2: the interaction between window size and relatedness index (figure 6.16, 0.39 R^2) and that between window direction and score (figure 6.15, 0.08 R^2). Indeed, performance patterns display little variation across the different parameter combinations, in particular for the most robust choices we identified through the inspection of the other, more explanatory, effects. As far as *relatedness index* is concerned, we already know that *neighbor rank* is the best choice, and from the interactions of window size with other parameters (transformation, score, and source corpus) we already know that the smaller windows ensure best performance. As for window direction, our intuition that parameters with a low explanatory power can be set to a default value (*undirected*, in this case) is confirmed in figure 6.15: as a matter of fact, there is little variation in the performance pattern of the different association measures, which is due to the manipulation of window direction. For the same reason, we set *criterion for context selection* to the more intuitive, easily interpretable and widely employed *frequency*.

Reduced setting Let us now turn to the discussion of the best parameter settings for the SVD-reduced runs.

The list of interactions in table 6.1 confirms the strong joint impact of *score* and *transformation* on model performance. The interaction is displayed in figure 6.17: the set of most robust values overlap with that of the unreduced runs: best results are achieved by association measures based on significance tests (*simple-ll, t-score, z-score*). SVD has, however, reshaped this interaction.

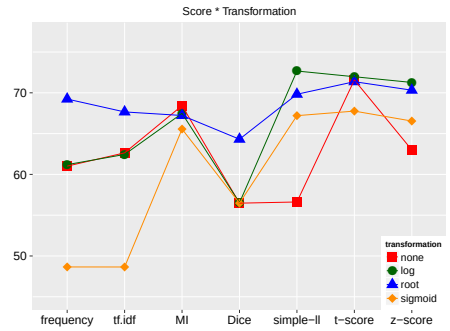

Figure 6.17: TOEFL, red: score / transformation

While in the unreduced setting the

best choice was *no transformation* for all measures but *simple-ll*, which in turn benefited from a *root* transformation, the picture has now shifted following a comparably systematic pattern: *simple-ll*, which exhibits the strongest variation in performance across different transformations, requires a more aggressive de-skewing (*log*); *z-score* and *t-score* require at least soft transformation (*root*); MI, for which *log* and *root* were detrimental in a reduced setting, is now robust to transformations. Overall, the general tendency seems to indicate that vector transformation in form of soft or aggressive deskewing is crucial to SVD performance. This tendency holds also for the weak values, as raw co-occurrence *frequency*, *tf.idf* and *Dice* only perform well in combination with a square *root* transformation, while in the unreduced setting feature transformation was not necessary.

The interaction between *feature score* and *distance metric*, displayed in figure 6.18. It shows that *cosine* is the best choice for all involved scores. While *simple-ll* is predicted, here, to be weaker than the other measures, we stick to it as our recommended score because the *feature score + transformation* interaction is a much stronger one in the feature ablation setting. The fact that *t-score* is predicted here as the strongest score suggests that it is a promising alternative to *simple-ll*.

The best *window size*, as shown in figure 6.19, is a 2-word window for all evaluated transformations. Even if the best window size is the same with respect to the unreduced runs, SVD reduction has reshaped this interaction, with respect to the picture sketched in figure 6.10: first, the detrimental effect of larger context windows is less marked (resulting in a lower position of the parameter in the feature ablation ranking); second, while *no transformation* was the most robust choice in the unreduced runs, SVD requires at least the soft de-skewing effect of a *root* transformation. As far as transformation is concerned, *root* and *log* are the best choices: this tendency is compatible with our picks for best scores (*simple-ll* and *t-score*).

The comparison between the score/transformation interaction (figure 6.17) and the window/transformation interaction (figure 6.19) aptly illustrates the dynamics of our linear modeling approach, and the potential shortcoming of our interpretation strategy: score/transformation prescribes *simple-ll* and *log*, while window/transformation indicates in *root* the most robust transformation across sizes. The latter finding is not surprising, and perfectly motivated based on interaction plot in figure 6.17: indeed, root transformation is the most robust across all scores, and this exactly the property which is captured in figure 6.19. A three-way modeling of such parameter clusters for which we observe strong, reciprocal interactions is the necessary next step.

Plot 6.20 displays the interaction between *feature score* and *source corpus*: more markedly than in the unreduced setting (cf. figure 6.13), association measures benefit from larger corpora: *UkWaC* is now the best choice for all scores but *Dice*. The choice of *number of context dimensions*, whose interaction with feature score is displayed in figure 6.21, is the same as for the unreduced runs: *5k* dimensions are sufficient. Differently than in the unreduced setting, however, a larger number of dimensions is detrimental for the strongest scores, with the exception of *simple-ll* which is still comparably robust also at 10k dimensions.*Frequency* and *tf-idf* are the most robust scores overall, exhibiting little variation in performance even up to 100k dimensions. A possible interpretative key for this interaction is the interplay between context dimensions and feature score in determining the sparsity of the space which SVD takes as an input. On the one hand, sparsity increases with a higher number of dimensions. On the other hand, the

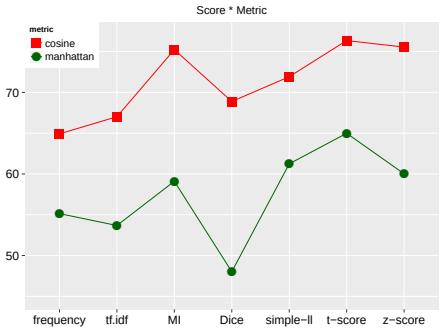

Figure 6.18: TOEFL, red: score / metric

Figure 6.19: TOEFL, red: window size / transformation

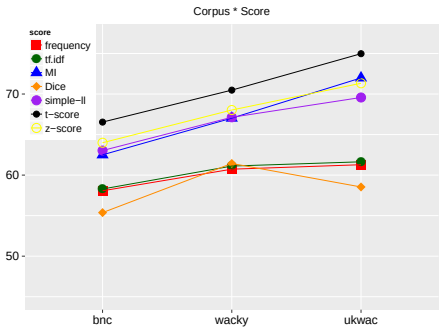

Figure 6.20: TOEFL, red, corpus / score

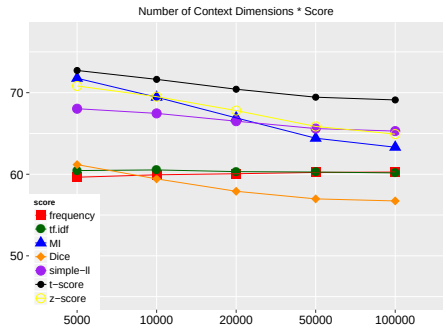

Figure 6.21: TOEFL, red, score / context.dim

frequency space is the densest in our experimental design, and we know that *tf.idf* and *Dice* do not increase sparsity with respect to frequency, while the other measures we employ do. The potential take-home message of this interaction could be that SVD dislikes extremely sparse spaces: a note of caution on this interpretation has to be made, however, as *Dice* exhibits a drop in performance. As far as *score* is concerned, its interactions with *corpus* and *number of context dimensions* confirm that *t-score* is a very strong alternative to *simple-ll*.

We conclude with a discussion of the dimensionality reduction parameters. The SVD parameters (*number of latent dimensions* and *number of skipped dimensions*) play a significant role in determining model performance. They show a tendency to participate in interactions with other parameters, but do not interact among themselves. We display the interaction between *metric* and *number of latent dimensions* in figure 6.22: the steep performance increase for both metrics shows that the widely-used choice of 300 latent dimensions Landauer & Dumais (1997) is suboptimal for the TOEFL task. The best value in our experiment is *900 latent dimensions*, and additional dimensions would probably lead to a further improvement.The interaction between *metric* and *number of skipped dimensions* is displayed in figure 6.23. While *manhattan* performs poorly no matter how many dimensions are skipped, *cosine* is positively affected by skipping 100 and (to a lesser extent) 50 dimensions. The latter trend has already been discussed by Bullinaria & Levy (2012).[2]

[2]The poor performance of *manhattan* distance in the SVD runs may be due to the fact that the vectors, normalized after SVD, have not been re-normalized with L1 norm after selecting the dimen-

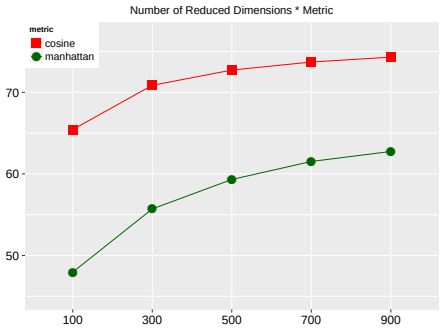

Figure 6.22: TOEFL, metric / red. dim.

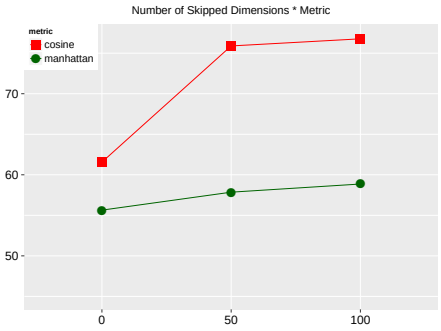

Figure 6.23: TOEFL, metric / skipped dim.

Given the minimal explanatory power of the *direction of the context window* and the *criterion for context selection* they are also set to their "unmarked" option: *undirected* and *frequency*. As discussed in the feature ablation section, the *index of distributional relatedness* has a very low explanatory power in this setting. *Neighbor rank* is the best choice, as anticipated in figure 6.14. However, consistently with the approach we adopted for the unreduced runs, given the small explanatory power of the parameter (i.e., the small difference between *rank* and *distance*), we recommend to employ *distance*.

Best settings In this section, we bridge the gap between DSM performance as predicted by our linear regression analysis, and actual performance. For unreduced and reduced runs, we report the performance of the best parameter settings set as discussed above, as well as the performance of three further DSM configurations which are picked based on different constraints (e.g., cognitive plausibility; comparison with the literature): the possibility of finding a robust DSM configuration by taking such constraints into account is a strong advantage of the methodology proposed in this thesis. Besides putting the evaluation results into a wider perspective, the discussion of the multiple best settings also provides a summarizing of the observations from the experiments. The state-of-the-art for TOEFL is already 100% accuracy by Bullinaria & Levy (2012). The performance of the best run is 87.5% for the unreduced experiments and 98.75% in the reduced experiments (see appendix B for the specific settings).

Tables 6.2 (unreduced runs) and 6.3 (reduced runs) display the following parameter settings, as well as their performance:

- *Best setting*: the best configuration based on the analysis of the effect plots;
- *Best cognitive*: a combination of parameter settings which is the best candidate as a "cognitive plausible" DSM representation. Such setting has *BNC* as a source corpus, because, compared to the larger corpora, it contains a more representative sample of language. Furthermore, it employs *frequency* and *log* transformation (a well-established corpus-based measure when it comes to psycholinguistic modeling) and *rank* as a relatedness index (because it allows for asymmetry). In the

sions. The lack of renormalization cannot affect cosine. Literature on DSM lacks a clear experimental account of the effect of (re)normalization, but the fact that *cosine* neatly outperforms *manhattan* in the unreduced runs indicate that our findings are reliable. Besides that, comparative experiments (*manhattan renormalized after SVD* vs. *manhattan normalized before SVD*) confirmed that renormalization improves *manhattan* performance, but this does not affect the overall behavior of other parameters.

reduced setting, the SVD parameters are set to a default value to make as few assumptions as possible concerning the cognitive interpretation of SVD and of the effects of its parameters:[3] the number of reduced dimensions is set to its maximum, 900 (given that almost uniformly across all our evaluation, more dimensions are never detrimental) and the number of skipped dimensions to zero (given its non homogeneous behavior across tasks, it is unlikely to be part of a mental representation). Remaining parameters (e.g., window size, number of context dimensions, distance metric) are set by inspecting the effect plots.

- *Best PPMI setting*: the closest parameter combination to the settings widely employed in the literature. Window size is set to 2: we considered this value to be the best approximation, within our experimental setup, of choices commonly made in the literature: while a window of size 1 is employed in some cases (e.g., by Bullinaria & Levy (2007) and Kiela & Clark (2014)), it is not clear whether the computation of window involves closed class words (but it is reasonable to assume it doesn't: in this case, this parameter would correspond to a window of size 2 in our setting); Baroni, Dinu, & Kruszewski (2014) employ a window of size 2. Furthermore, we have *MI* and *no transformation* (corresponding to PPMI), the largest corpus (*UkWaC*) and the largest set of contexts *100k* (large scale studies usually employ even larger corpora and context sets, so this is the best approximation in our parameter set), *cosine* and *distance* (as rank is evaluated for the first time in this thesis). Following on the recommendations of Bullinaria & Levy (2012), we set the number of latent and skipped dimensions to their maximum (900 and 100, respectively). Window size is a task-specific parameter, and thus set by inspecting the effect plots.

- *Best PPMI+ setting*: we identify a DSM setting which is based on *MI* and *no transformation*, but set the remaining parameters with our evaluation methodology The goal here is to check whether, thanks to our large-scale experiments and to the full picture of parameter interactions gathered with our regression analysis, we can find improvements to the widely employed PPMI setting. Note that the other parameters might be substantially different from the "best PPMI" setting.

setting	corpus	win	direction	c.dim	exc	score	transf	metric	rel.ind	accuracy
Best setting	ukwac	2	undir	10000	f	z-score	none	cosine	rank	81.25
Best cognitive	bnc	1	undir	100000	f	frequency	log	cosine	rank	75.00
Best PPMI	ukwac	2	undir	100000	f	MI	none	cosine	dist	76.25
Best PPMI+	ukwac	2	undir	5000	f	MI	none	cosine	rank	82.50

Table 6.2: TOEFL, unreduced – best settings

Based on the performance of the DSM settings in table 6.2 and 6.3, we can draw the following conclusions:

- SVD improves performance on the TOEFL task, as the best robust setting for the reduced runs has a better performance than the best robust setting in the unreduced runs.

[3]Besides the claims made in Landauer & Dumais (1997), there is no evidence for a cognitive plausibility of SVD, nor for the 300 dimensions employed in that foundational SVD paper to be the optimal value for such parameter.

setting	corpus	win	direction	c.dim	exc	score	transf	n.dim	dim.skip	metric	rel.ind	accuracy
Best setting	ukwac	2	undir	5000	f	simple-ll	log	900	100	cosine	dist	93.75
Best cognitive	bnc	1	undir	100000	f	frequency	log	900	0	cosine	rank	70.00
Best PPMI	ukwac	2	undir	100000	f	MI	none	900	100	cosine	dist	88.75
Best PPMI+	ukwac	8	undir	5000	f	MI	none	900	100	cosine	dist	91.25

Table 6.3: TOEFL, reduced – best settings

- This trend is reversed in the cognitively inspired setting: we abstain from "cognitive" interpretations on this fact, and prefer to go for a mathematical one assuming that the information gathered from BNC at 100k dimensions is just too sparse (and, possibly, looking at the effect plot in figure 6.21 just not sufficient given the minimal improvement with respect to smaller number of contexts), and log transformation too aggressive, to ensure a good dimensionality reduction. As a matter of fact, our evaluation would predict a root transformation to be the best choice. Moreover, the lack of improvement with SVD may be due to the fact that, given the lack of reasonable cognitive assumptions concerning the SVD parameters, we adopt here an untuned SVD.

- Our methodology, combined with the thorough exploration of the parameter space we conducted, allows us to further improve PPMI with respect to the settings which are commonly employed in the literature (*Best PPMI+* better than *Best PPMI*, in both settings). This is not only because we experiment with rank, which improves performance in the unreduced setting. Contrary to what is commonly done in the literature, we found that in both reduced and unreduced setting PPMI, at least on the TOEFL task, benefits from a smaller set of contexts; moreover, in the reduced setting, it benefits from a window which is larger than commonly assumed. Note that TOEFL is a fairly small dataset, so in the next sections we will check whether such observations concerning the interaction of PPMI with other parameters generalize also to the other datasets.

6.2 Similarity ratings

Tested in the task of predicting the unsigned Pearson correlation between DSM similarities and human similarity ratings, the linear models achieve the following adjusted R^2 values: for RG65, 91% (unreduced) and 86% (reduced); for WS353, 94% (unreduced) and 90% (reduced). The model fit is higher for WS353 than it is for RG65, showing that the linear models have been able to build a more robust generalization for the former: this is not surprising, given that WS353 has a much larger number of items (353 vs. 65, cf. TOEFL, 88 items: 87% unreduced, 89% reduced) and performance on smaller datasets is more likely to fluctuate due to noise.

The plots in 6.24 display the ranking of the evaluated parameters according to their importance in the feature ablation setting, for the unreduced (left) and reduced (right) runs. The tables in 6.4 display the interactions which account for more than 0.5% of explained variance, in the unreduced (left) and reduced (right) setting, respectively.

Similarly to what we already observed for TOEFL, *feature score* and *feature transformation* appear to play a crucial role in determining DSM performance, both in the reduced and in the unreduced setting. The tables in 6.4 show that their interaction is the strongest for both tasks, in the reduced and unreduced setting.

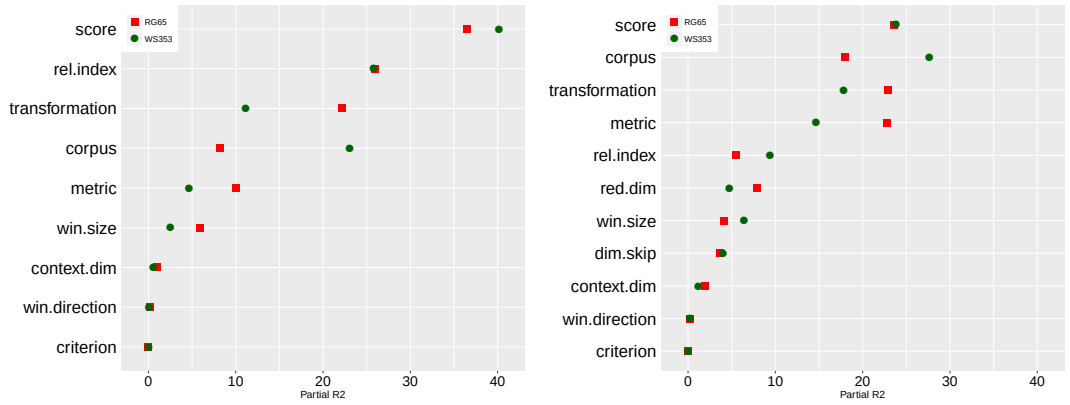

Figure 6.24: Similarity Ratings: Feature ablation. Left: unreduced; Right: reduced.

Interaction	df	RG65	WS353
score:transf	18	8.40	4.39
score:rel.index	6	2.37	3.24
corpus:score	12	0.80	1.75
transf:metric	3	1.66	0.55
win.size:score	24	0.76	1.02
win.size:transf	12	1.08	0.57
score:metric	6	0.85	—
win.size:rel.index	4	0.69	—

Interaction	df	RG65	WS353
score:transf	18	10.28	8.66
metric:red.dim	4	2.18	1.42
score:metric	6	1.91	0.59
win.size:transf	12	1.43	1.01
corpus:metric	2	1.83	0.51
metric:context.dim	4	1.08	0.62
corpus:score	12	0.77	0.82
score:dim.skip	12	0.58	0.85
win.size:score	24	0.77	0.69
win.size:metric	4	—	0.65
transf:dim.skip	6	—	0.54
metric:dim.skip	2	—	0.82
score:context.dim	24	0.56	—

Table 6.4: Ratings: interactions, partial R^2. Unreduced (left) vs. Reduced (right)

Index of distributional relatedness plays a major role in the unreduced runs, and, comparably to TOEFL, loses explanatory power in a reduced setting. Yet, there are two main differences in the behavior of this parameter with respect to the multiple choice task. First, the parameter is much stronger in the unreduced setting, where it also participates in a strong interaction with *feature score*. Second, while with TOEFL SVD almost completely neutralized the difference between *distance* and *rank*, their difference is here still strong, both for RG65 and WS353.

The loss in explanatory power of *relatedness index* in the reduced setting is accompanied by a raise in feature ablation by *corpus* and *metric*. The interaction tables show that such gain is only partially due to interactions, which mainly affect *metric* and are quite weak anyway. This indicates that the best parameter values of such parameters are robust across the possible combinations with the values of other parameters.

Feature ablation proves *size of the context window* to be a rather weak parameter, both in the reduced and unreduced setting: interestingly, while the parameter affects performance on RG65 more than it does on WS353 in the unreduced runs, it turns out to be more crucial to WS353 in the reduced runs. In our exploration of best parameter values we will elaborate more on the interpretation of this effect.

The two SVD-related parameters are less powerful in this task than they were in TOEFL: in particular *number of skipped dimensions*, among the top-ranked in the

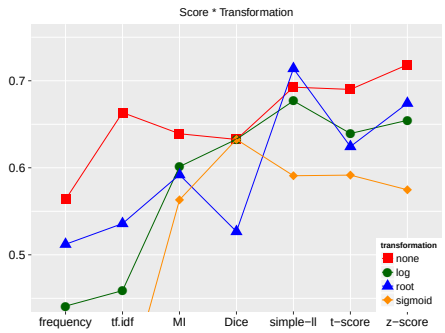

Figure 6.25: RG65, unred, score / transformation

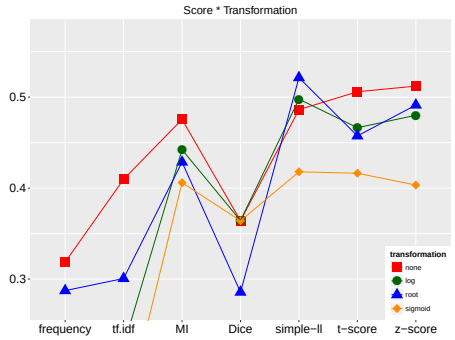

Figure 6.26: WS353, unred, score / transformation

multiple-choice synonymy task, has here a rather weak explanatory power.

Number of context dimensions has a very weak explanatory power, and it interacts with other parameters only in a reduced setting. Similarly for what we observed for TOEFL, *exclusion criterion* and *direction of the context window* have a neglectable impact on DSM performance, and they do not participate in any interaction.

6.2.1 Best parameter values

Unreduced setting In what follows, we discuss the best parameters for the RG65 and WS353 in the unreduced experiments. We follow the ranking of interactions in table 6.4, and display the effect plots for the two datasets side by side. Note that, given the different range of performance (RG65 is much easier than WS353), producing plots on the same scale would have been unfortunate at the level of visualization. As already discussed for TOEFL, what matters for the purpose of this discussion are *differences in predicted performance* due to manipulation of our parameters, and the differences on the y-axis of the displayed plots are equivalent.

We start from the top of the interaction table, where we find, once again, the interaction between *feature score and feature transformation* displayed in figure 6.25 and 6.26. The pattern is quite similar for the two tasks, and strikingly similar to what we already saw for TOEFL: *simple-ll, t-score* and *z-score* are the best performing measures, and *no transformation* is the best choice for all involved scores, with the exception of *simple-ll*, which requires the soft de-skewing of a *root* transformation. More specifically, the best choices for both datasets are *z-score* without transformation, or, alternatively, *simple-ll* with *root* transformation, with a preference for the latter in WS353. The interaction between *feature score and relatedness index*, in figures 6.27 and 6.28 indicates unambiguously that the best relatedness index is *rank*, and confirms *simple-ll* and *z-score* as the strongest scores.

The joint inspection of the interactions between *window* and *feature score* (figures 6.29 and 6.30) and *feature transformation* (figures 6.31 and 6.32) uncovers the only substantial difference between the two tasks, namely the best parameter value for *window size*. As a general pattern, WS353 needs larger context windows than RG65 does: one possible explanation for this observation is the different composition of the WS353 dataset, which includes examples of semantic relatedness beyond attributional similarity. In more detail, we observe that if *simple-ll* and *root* are kept as the main choices for transformation, a 2-word window is sufficient for RG65 while an 8-word is necessary

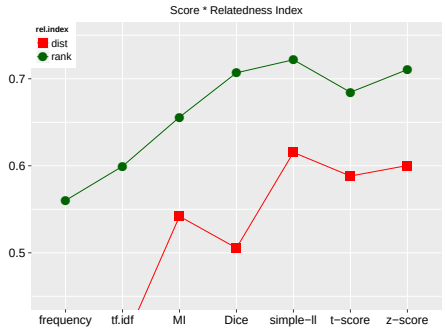

Figure 6.27: RG65, unred, score / rel.index

Figure 6.28: WS353, unred, score / rel.index

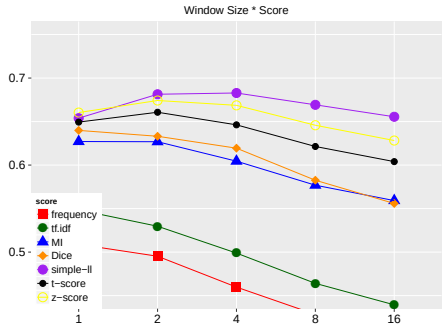

Figure 6.29: RG65, unred, window / score

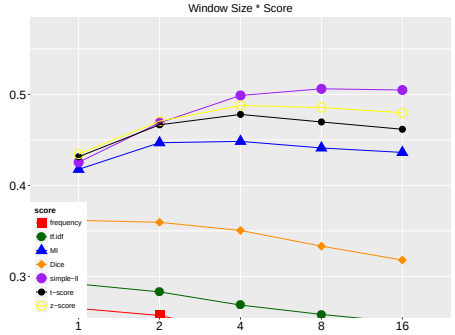

Figure 6.30: WS353, unred, window / score

for WS353. If, as an alternative, we would go for the *z-score* and *no transformation combination*, slightly larger context windows (4 for RG65 and 8/16 for WS353) would be necessary, with a 4-word window as a reasonable compromise if in need to establish a shared best setting for both datasets.

The interaction between *score* and *source corpus* allows us to set a shared best parameter for both datasets: *WaCkypedia*, suggesting that this task benefits from a trade-off between quality and quantity (WaCkypedia being smaller and cleaner than ukWaC, but less balanced than the BNC); we can already anticipate that this pattern will characterize the reduced runs, as well. Note that, due to the presence of neighbor rank in our experimental setup, and given that, differently from distance, rank operates over the distributional representations of the whole vocabulary (vs. distance, for which only the vectors of the items in the dataset are needed), it would be simplistic (and probably wrong) to interpret this result as due to the fact that the items in the ratings datasets are better represented by the *WaCkypedia* contexts. From this perspective, *neighbor rank* represents a less task-dependent oriented measure of DSM performance.

The interaction between *feature transformation* and *distance metric*, not displayed here for reasons of space (plots are available in the supplementary material) follows a clear pattern for both datasets: for all evalutated transformations, *cosine* is the best choice, with a large advantage over *manhattan* in all cases but *no transformation* (cf. TOEFL, where we observed the very same pattern).

The two RG65-specific interactions (window size and relatedness index; score and metric), are not discussed here for reason of space and because they do not affect the

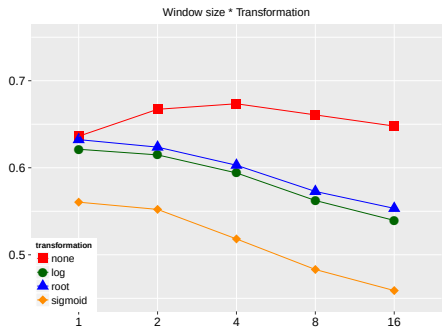

Figure 6.31: RG65, unred, window / transformation

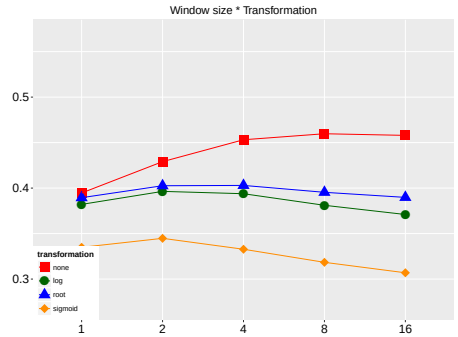

Figure 6.32: WS353, unred, window / transformation

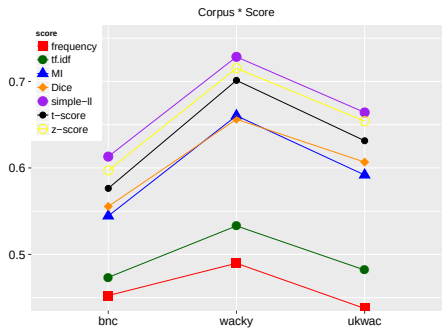

Figure 6.33: RG65, unred, corpus / score

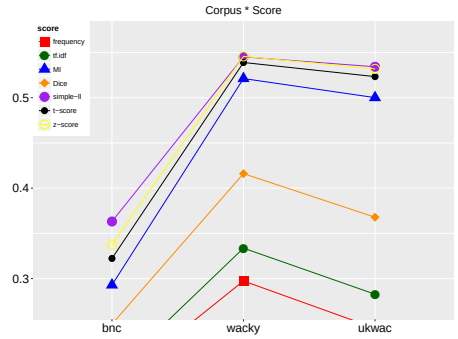

Figure 6.34: WS353, unred, corpus / score

choice of best parameters values we already made, based on the stronger interactions.

For *number of context dimensions*, which has a very weak explanatory power, we resort to the main effect (see supplementary material) and pick a matrix of *20k* dimensions. Consistently with the approach adopted for TOEFL, we set the remaining parameters ($R^2 < 0.5$) to a default value: *exclusion criterion* to *frequency* and *direction* to *undirected*.

Summing up, the best model configuration in an unreduced setting is *WaCkypedia* as a source corpus, *20k* context dimensions based on *frequency*, an *undirected* window of size *2* for RG65 and *8* for WS353, *simple-ll* and *root* transformation, *cosine* distance and *neighbor rank*.

Reduced setting In what follows, we discuss the best parameter settings for the runs involving dimensionality reduction, taking a comparative approach to what we already discussed for the unreduced runs, as well as the trends identified for TOEFL. Our starting point is, as in the previous section, the ranking of interactions listed in table 6.4.

The interaction between *score* and *transformation* is confirmed as the strongest in terms of explanatory power. With respect to the unreduced setting, the interaction has undergone a shift similar to what we already identified for TOEFL: while before SVD a soft transformation (*root*) was necessary only for *simple-ll*, while the other measures achieved their best performances without any transformation, *root* has become the best choice for nearly all scores (with the only exception of *MI*, which reaches its best

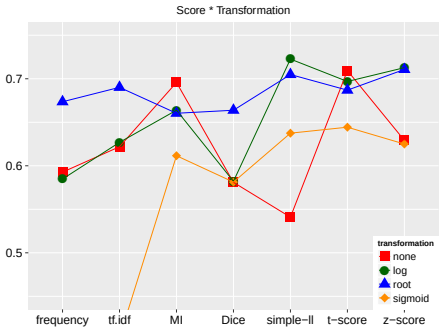

Figure 6.35: RG65, red, score / transformation

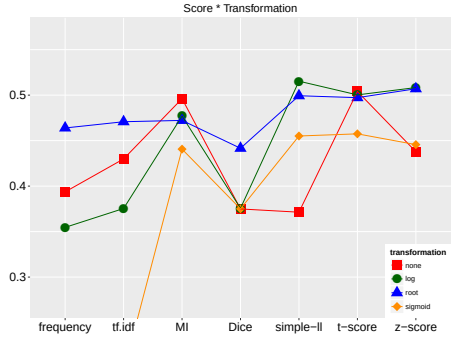

Figure 6.36: WS353, red, score / transformation

Figure 6.37: RG65, red, window / transformation

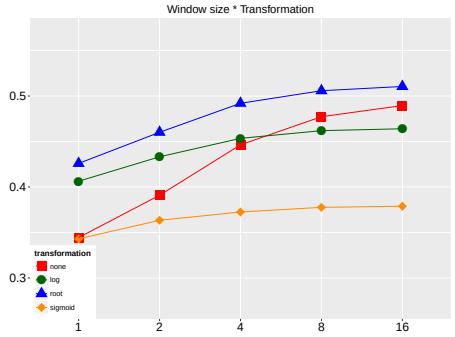

Figure 6.38: WS353, red, window / transformation

predicted performance without any transformation) and *simple-ll* requires a strongest de-skewing, *log*. The inspection of the plots displaying the interaction between *score* and *metric* confirms that *cosine* is the best measure and does not affect the choice of best values for *score*, hence we do not discuss them here and we refer the reader to the supplementary material.

The choice of the optimal *window size* depends on *transformation*, as shown in figures 6.37 and 6.38 for RG65 and WS353, respectively. Once again, the way SVD reshapes the interaction is comparable to what we observed for TOEFL, as larger windows improve performance, or at least are not detrimental as they used to be in a reduced setting. On the RG65 dataset, figure 6.37 shows that for a *logarithmic* transformation – which we already identified as the best *transformation* in combination with significance association measures – the highest performance is achieved with a *4 word window*. *Root* transformation, which is the best for the other measures, displays the same pattern. *No transformation* which, before SVD, exhibited a drop after the intermediate values, keeps on improving up to a 16 word window. The corresponding effect display for WS353 (figure 6.38), which before SVD was exhibiting a preference for windows larger than RG65, has followed the same pattern: now all transformations increase (or are at least unaffected) up to 16, while *simple-ll* stabilizes at an *8 word window*. A *4-word window* is confirmed as a robust choice for both datasets. The interaction between *window size* and *score*, not displayed here for space constraints, confirms the tendencies discussed so far.

To identify the best *corpus*, we jointly inspect its interaction with *metric*, in figures

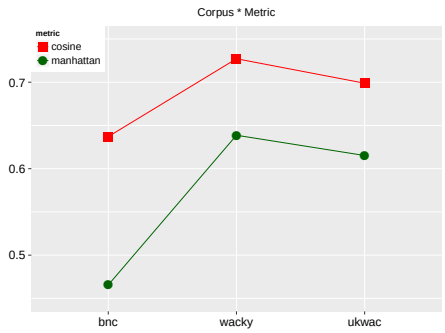

Figure 6.39: RG65, red, corpus / metric

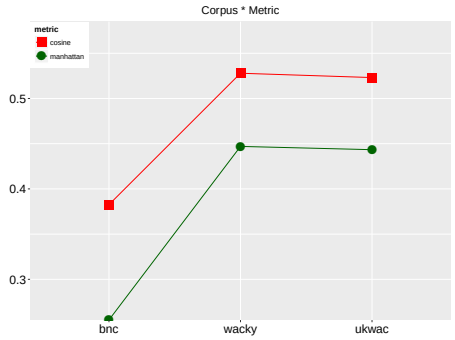

Figure 6.40: WS353, red, corpus / metric

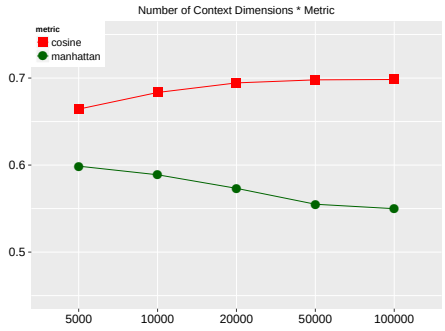

Figure 6.41: RG65, red, metric / context dim

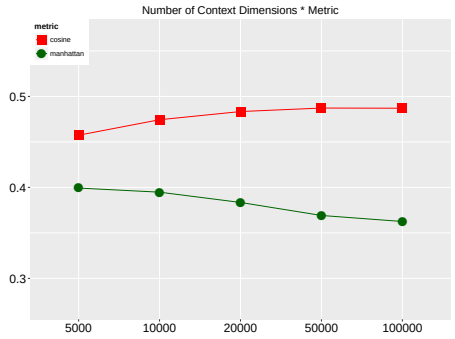

Figure 6.42: WS353, red, metric / context dim

6.39 and 6.40. *WaCkypedia* is confirmed as the best source corpus for both datasets, at the same level of *UkWaC* for WS353. The interaction between *corpus* and *score*, not shown here, confirms this tendency. Needless to say, *cosine* is the best distance.

The interaction between *metric* and *number of context dimensions* (figure 6.41 and 6.42) confirms *cosine* as the best distance metric, and it allows us to identify in *50k* and *20k* the best values for the number of context dimensions, for both datasets. It is also a very good example of how crucial it is to take into account interactions, as we see that the two metrics display an almost specular pattern, which would have been averaged away in the corresponding main effect (refer to the main effect plots in the supplementary material).

Let us now turn to the dimensionality reduction parameters. The interaction between *number of latent dimensions* and distance *metric* was the second strongest, overall, in the feature ablation setting. Figures 6.42 and 6.43 show that *cosine* outperforms *manhattan* across the board, and that with cosine as a distance metric and at least *300 latent dimensions* (up to 500 for WS353) are the best choice, and no further improvement is to be expected at larger dimensionalities.

To learn more about the best value of *number of skipped dimensions* we inspect its interaction with *score* (figures 6.45 and 6.46). For the best measures, the best results are predicted with *50 skipped dimensions*. For RG65, even skipping up to 100 dimensions does not affect particularly the performance, for the strongest association measures. For WS353, on the other hand, skipping 100 dimensions is clearly detrimental, and to select 50 as our final we had to consult the (weaker) interactions with *metric* and

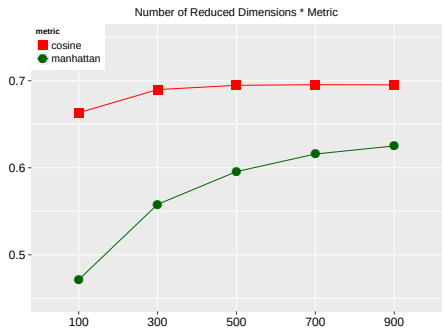

Figure 6.43: RG65, metric / red.dim

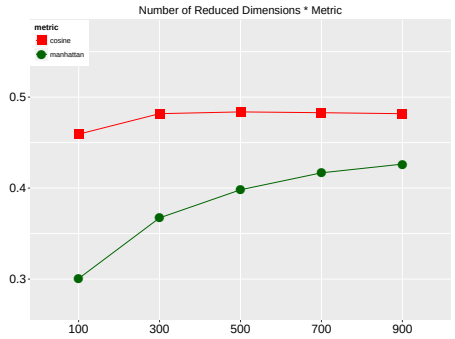

Figure 6.44: WS353, metric / red.dim

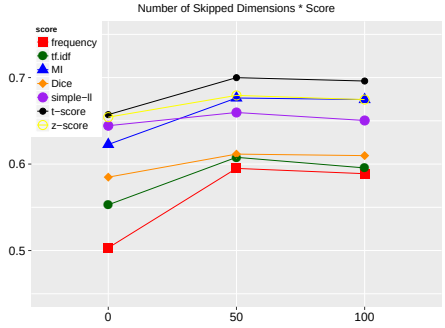

Figure 6.45: RG65, score / skipped dim

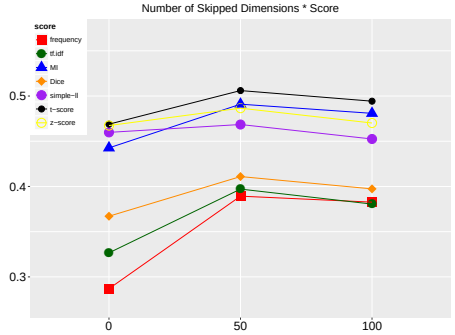

Figure 6.46: WS353, transformation / skipped dim

transformation, not shown here for reasons of space. The preference of RG65 for 50/100 skipped dimensions compared to 0/50 of WS353 can be interpreted in light of the different type of relations between the items in the two datasets: as already discussed, WS353 encodes both similarity and topical relatedness. As a matter of fact, RG65 patterns with TOEFL more than WS353 does. We will come back to this point later, in chapter 8, where a direct comparison of paradigmatic (similarity) and syntagmatic (relatedness) will be carried out in the same parameter space.

Differently from what happened in the TOEFL task, *relatedness index* still plays a strong role in the reduced setting. Best choice is, again, *neighbor rank*: see section 6.4 for an extended discussion of this effect. For *exclusion criterion* and *window direction* we stick to the same default choices for best parameters.

Summing up, our investigation has identified the following best choices: *WaCkypedia* as corpus, an *undirected window* of *4 words* for RG65 and *8/16 words* for WS353, *50k* context dimensions selected based on *frequency*, *simple-ll* with a *log* transformation, *500* reduced dimensions skipping the *first 50 dimensions*, *cosine* distance and *neighbor rank*.

Best settings In what follows, we summarize our observations on the rating datasets by displaying the performance of the best settings for the task of predicting the similarity ratings, and compare such best settings to a cognitive plausible DSM and to two PPMI settings, with parameters picked as described in section 6.1.1; the only difference is that, here, we take 500 latent dimensions to be best PPMI setting (following Baroni & Lenci (2010) and Levy et al. (2015)). Table 6.5 and 6.6 summarize the observations

discussed in the previous sections and display both Pearson's r and also Spearman's *rho*, for completeness and better comparison with the literature.

The state-of-the-art of corpus-based systems[4] is 0.86 r (0.83 *rho*) for RG65 (Hassan & Mihalcea, 2011) and 0.81 *rho* for WS353 (Halawi et al., 2012). For the performance of the best runs and the specific settings, see appendix B.

The performances of the DSMs displayed in the two tables allows us to draw the following conclusions:

- The regression methodology allowed us to identify robust best settings which are, for RG65, better than the state-of-the-art of comparable DSMs.
- Comparably to what we observed with TOEFL, SVD improves DSM performance: for both datasets, robust best settings in the reduced runs have better performances than in their unreduced counterpart.
- As in TOEFL, the cognitive setting performs rather poorly and is negatively affected by SVD reduction. This is particularly true for WS353: it is reasonable to assume that the experimental items for this dataset just have poorer representations in the BNC. As for the negative impact of SVD, note that, as in TOEFL, in the cognitive setting we do not to manipulate a crucial parameter, i.e., the number of skipped dimensions.
- For both datasets and in both reduced and unreduced runs, the best PPMI+ models outperform the corresponding PPMI counterparts, showing that the use of neighbor rank (which hasn't yet been tested in the predictions of similarity ratings) and the possibility of setting other parameters with respect to their interactions to PPMI does improve DSM results.

setting	corpus	win	direction	c.dim	exc	score	transf	metric	rel.ind	r	rho
RG65											
Best setting	wacky	2	undirected	20000	f	simple-ll	root	cosine	rank	**0.82**	**0.83**
Best cognitive	bnc	1	undirected	20000	f	frequency	log	cosine	rank	0.66	0.67
Best PPMI	ukwac	2	undirected	100000	f	MI	none	cosine	dist	0.67	0.72
Best PPMI+	wacky	2	undirected	20000	f	MI	none	cosine	rank	0.81	0.81
WS353											
Best setting	wacky	4	undirected	20000	f	simple-ll	root	cosine	rank	**0.67**	**0.69**
Best cognitive	bnc	2	undirected	20000	f	frequency	log	cosine	rank	0.36	0.37
Best PPMI	ukwac	2	undirected	100000	f	MI	none	cosine	dist	0.57	0.60
Best PPMI+	wacky	2	undirected	20000	f	MI	none	cosine	rank	0.63	0.65

Table 6.5: Ratings, unreduced – best settings

[4]Systems which are classified as knowledge-based or hybrid in the ACL state-of-the-art wiki (`https://aclweb.org/aclwiki/State_of_the_art`) are not directly comparable to the ones discussed in this thesis. Such systems hold state-of-the-art performance for both RG65 (Pilehvar & Navigli (2015): 0.92 *rho*, 0.91 r) and WS353 (Speer et al. (2017), 0.83 *rho*).

setting	corpus	win	direction	c.dim	exc	score	transf	n.dim	dim.skip	metric	rel.ind	r	rho
RG65													
Best setting	wacky	4	undirected	50000	f	simple-ll	log	500	50	cosine	rank	**0.87**	**0.85**
Best cognitive	bnc	4	undirected	100000	f	frequency	log	900	0	cosine	rank	0.60	0.59
Best PPMI	ukwac	2	undirected	100000	f	MI	none	900	0	cosine	dist	0.76	0.77
Best PPMI+	wacky	8	undirected	10000	f	MI	none	500	50	cosine	rank	0.83	0.77
WS353													
Best setting	wacky	16	undirected	50000	f	simple-ll	log	500	50	cosine	rank	**0.69**	**0.71**
Best cognitive	bnc	8	undirected	50000	f	frequency	log	900	0	cosine	rank	0.30	0.29
BestP PMI	ukwac	2	undirected	100000	f	MI	none	900	0	cosine	dist	0.57	0.60
Best PPMI+	wacky	16	undirected	50000	f	MI	none	500	50	cosine	rank	**0.69**	**0.71**

Table 6.6: Ratings, reduced – best settings

6.3 Clustering

The feature ablation plots in figure 6.47 display the importance of the evaluated parameters in the clustering task (adj. R^2, unreduced: AP: 86.1%; BATTIG: 79.7%; ESSLLI: 68.4%; MITCHELL: 82.9%. adj. R^2, reduced: AP: 82.0%; BATTIG: 77.0%; ESSLLI: 58.1%; MITCHELL: 73.3%).[5] Parameter ranking is determined by the average of the feature ablation values over all four datasets. The tables in 6.7 reports all parameter interactions that explain more than 0.5% of the total variance for each of the four datasets.

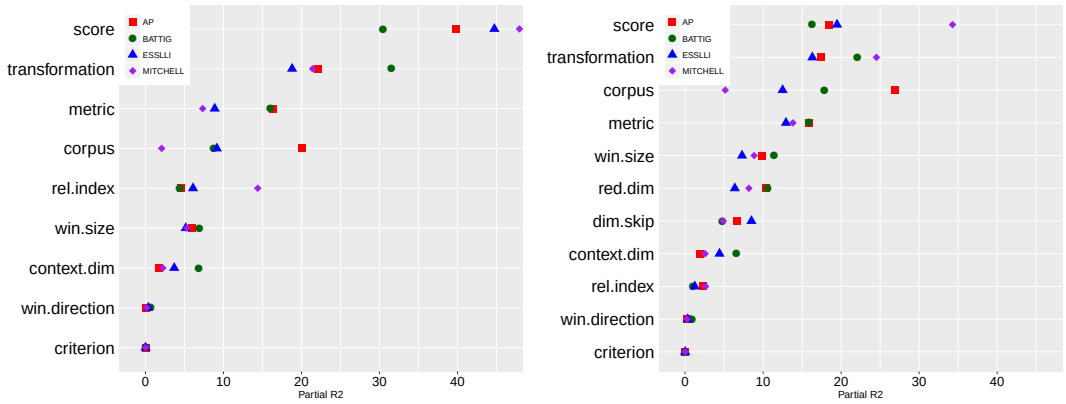

Figure 6.47: Clustering: Feature ablation. Left: unreduced; Right: reduced.

The feature ablation picture for the clustering datasets follows the common patterns already observed for TOEFL, RG65 and WS353: *score* and *transformation* play the strongest role in determining model performance, both in the unreduced and reduced setting. *Distance metric* and *source corpus* follow in the feature ablation ranking, the latter gaining explanatory power with SVD (a trend we already observed for the similarity ratings), and displaying a wide variation in explanatory power across the different datasets.

Relatedness index, rather powerful in the unreduced setting displays a feature ablation loss with SVD, confirming the trend we observed for the other datasets. Yet another familiar trend is displayed by *window size*: comparably to the other datasets, this parameter sees its impact on model performance increase when SVD is applied.

[5]The low R^2 of ESSLLI is explained by the very small size of the dataset, which contains 44 items, compared to AP (402), BATTIG (80), and MITCHELL (60).

Interaction	AP	BAT	ESS	MIT
score:transf	6.02	8.98	9.02	7.13
transf:metric	5.64	4.51	3.66	1.79
win.size:transf	1.81	3.38	1.03	1.30
score:metric	2.88	1.63	1.42	0.78
score:rel.index	1.88	0.55	1.99	1.71
win.size:score	1.08	0.98	1.87	0.63
score:cont.dim	0.98	0.54	1.20	0.61
corpus:metric	0.76	0.53	2.18	-
corpus:score	1.16	-	1.40	-
corpus:cont.dim	-	0.59	1.00	-
corpus:transf	-	-	1.09	-
win.size:metric	-	0.73	-	-

Interaction	AP	BAT	ESS	MIT
score:transf	7.10	7.95	7.56	11.42
metric:red.dim	3.29	3.16	2.03	2.03
win.size:metric	2.22	1.26	2.97	2.72
win.size:transf	2.00	2.95	0.88	2.66
corpus:metric	1.42	2.91	2.79	1.11
metric:dim.skip	2.25	1.54	2.77	0.86
corpus:win.size	2.36	1.18	1.49	1.23
score:dim.skip	0.56	1.15	0.99	1.39
win.size:score	0.74	0.77	0.54	0.65
metric:cont.dim	-	1.20	0.67	0.92
transf:dim.skip	-	1.17	0.85	1.39
transf:metric	-	-	1.17	1.00
corpus:cont.dim	-	0.70	1.25	-
corpus:transf	0.66	0.56	-	-
corpus:red.dim	-	-	0.99	-
corpus:score	-	-	0.61	-
metric:rel.index	-	-	-	0.58
score:cont.dim	-	-	-	0.52
score:metric	0.56	-	-	-

Table 6.7: Clustering (AP, BAT[TIG], MIT[CHELL], ESS[LLI]): interactions, R^2. Unreduced (left) vs. Reduced (right)

Both in the unreduced and reduced setting, the *number of context dimensions* is only weakly influential in determining model performance: also this pattern is one we already observed in the previous discussions throughout this chapter; it confirms that, if other parameters are properly set, they have the potential neutralize the impact of computationally impactful choices like the size of the context matrix.

Dimensionality reduction parameters, *number of latent dimensions* and *number of reduced dimensions* are both strongly influential.

The interaction tables displayed in 6.7 allow us to identify a core of interactions which are common to all datasets: they involve *score*, *transformation* (in either reduced and unreduced setting, *score and transformation* is the strongest interaction), and *metric*; additionally, in the unreduced setting, the core interactions involve *window size*, as well; in the reduced setting, the SVD parameters participate in the set of core interactions.

The main notable differences between reduced and unreduced setting is the interaction between *score* and *relatedness index* in the former; interactions of the strongest parameters with *corpus* and *number of context dimensions* tend to involve only a subset of the datasets in the unreduced setting and enter the set of core interactions only marginally.

6.3.1 Best parameter values

Unreduced setting The interaction between *score* and *transformation* is the strongest across all the clustering datasets, and, as shown in figures 6.48 to 6.51, displays comparable dynamics for the four datasets, also in line with the tendencies we identified for TOEFL, RG65 and WS353. The best combination is *simple-ll* and *root*, and, again, *no transformation* is the best choice for all the other scores, with the only exception of ESSLLI, for which best transformations are *simple-ll+log* and *z-score+root*.

In the following discussion, we focus on the AP dataset, which is larger and thus

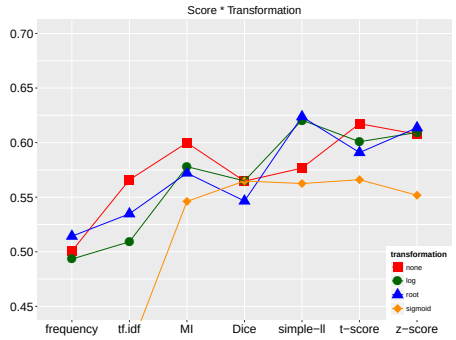

Figure 6.48: AP, unred, score / transformation

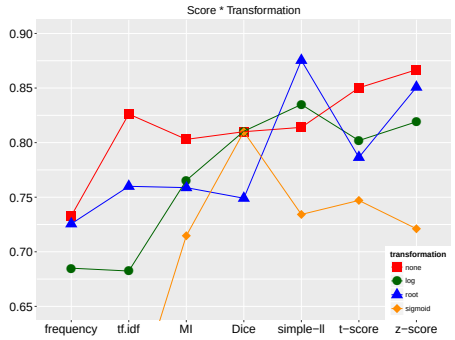

Figure 6.49: BATTIG, unred, score / transformation

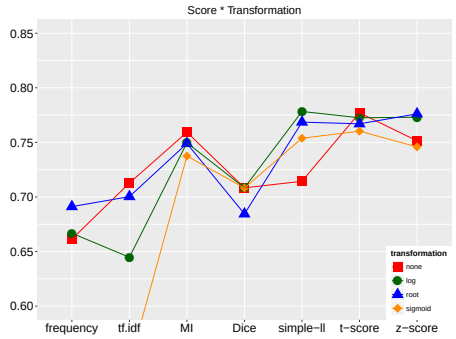

Figure 6.50: ESSLLI, unred, score / transformation

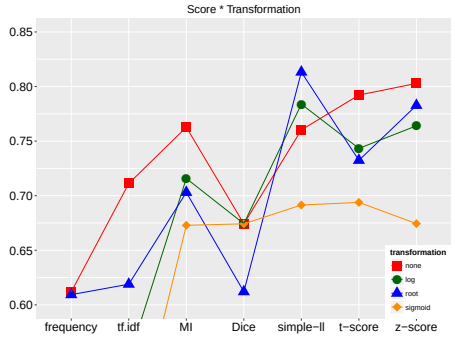

Figure 6.51: MITCHELL, unred, score / transformation

more reliable than the other three datasets. We mention remarkable differences between the datasets in terms of best parameter values. For more detailed comparisons, we refer the reader to the full plots displayed in the supplementary material; for a full overview of the best parameter setting for each dataset, see table 6.8.

Let us set the best parameter value for *window size*. As it turns out, we once again face the need to integrate information coming from different interactions. The interaction between *window* and *transformation* is displayed in figure for 6.52 for AP. For *log* and *root* transformation, a very small context window (one or two words) is the best choice. *No transformation*, on the other hand, keeps on increasing its predicted performance up to at least a 4-word window. In contrast, the interaction between *window size* and *score* (figure 6.53) identifies in a 4-word window the best parameter choice for *simple-ll*. ESSLLI and MITCHELL follow a comparable pattern: the best window size value in combination with the best feature/score is a *2 or 4 word window*. Figures 6.54 and 6.55 illustrate the same pair of interactions for BATTIG: here, it is straightforward to make a choice as the two effects do not contradict each other and a *4 word window* appears to be the obvious choice.

Our discussion now turns to the interactions between *score* and *transformation* with *relatedness index* and *metric*, respectively. The interaction between *feature score* and *relatedness index* is displayed in figure 6.56. For *simple-ll* (our best score for AP), *neighbor rank* outperform their competitors, albeit of a small margin (the interaction between score and metric, not shown here, confirms the preference for cosine). The same choices can straightforwardly be applied to BATTIG and MITCHELL. For ESSLLI,

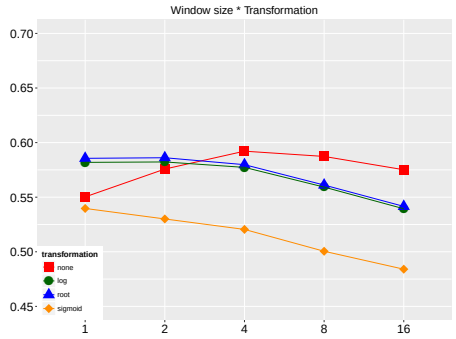

Figure 6.52: AP, unred, window / transformation

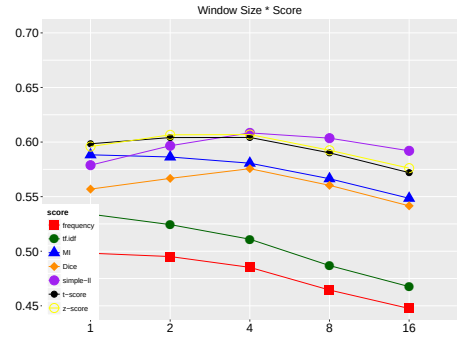

Figure 6.53: AP, unred, window / score

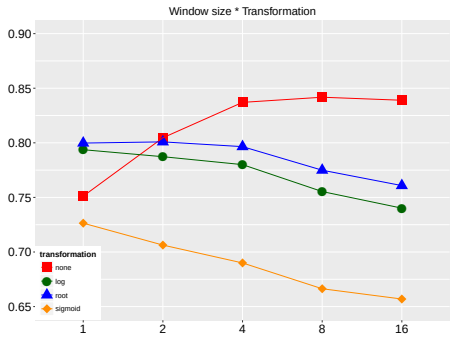

Figure 6.54: BATTIG, unred, window / transformation

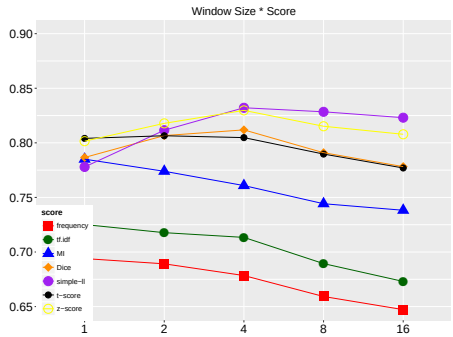

Figure 6.55: BATTIG, unred, window / score

however, the choice of distance metric is more complicated, as effects plots reveal a contradictory picture (in some interactions *cosine* is the best choice, in some others *manhattan* wins). As a matter of fact, it is possible to find a best setting based on either distance metric: here, we stick to *cosine* because, given the small size of this dataset, we tend to trust more the tendency that has shown as predominant in the rest of the experiments.

Figure 6.57 displays the interaction between *feature transformation* and *distance metric*, for the AP dataset: *cosine* outperforms *manhattan* in combination with all transformations, but is outperformed by it when no transformation is applied. All the other clustering datasets show a comparable pattern. Interestingly, for the other tasks, we observed comparable tendencies, with *cosine* outperforming *manhattan* with all transformations, but being substantially equivalent to it without any transformations. This result confirm how crucial it was to bring vector transformation into the picture, while other evaluation studies, even when targeting different distance measures/metrics (or different feature scores), did not consider its role (besides positivization or shifting by a global costant – see discussion in section 2.3.3.1).

The exploration of the remaining interactions, whose plots are not shown here for reasons of space, allows to make the following parameter choices: the best *number of context dimensions* is *20k* for AP, *50k* for MITCHELL, and *100k* for BATTIG and ESSLLI. The best *source corpus* is *WaCkypedia*.

The discussion of best parameter values has highlighted some very crucial points of our interpretation methodology: the need to integrate the information coming from

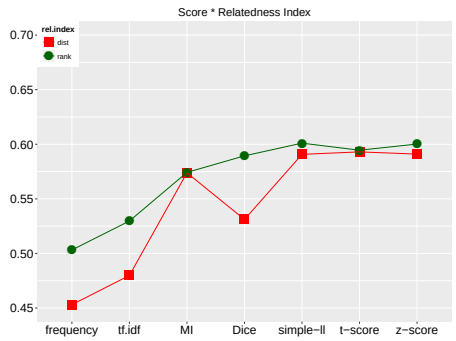

Figure 6.56: AP, unred, score / relatedness index

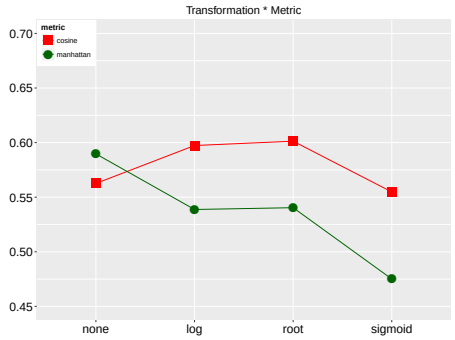

Figure 6.57: AP, unred, tranformation / metric

different interactions (which shouldn't be dealt with in a completely automatic fashion, but based on qualitative considerations); the potential issues arising from very small datasets such as ESSLLI: while selecting best parameter values, we highlighted its somehow idiosyncratic behavior with respect to the other datasets (e.g., its weak preference for *manhattan*). As already pointed out, such idiosyncratic trends need to be taken with a note of caution because they could be an instance of a higher level of overfitting in the performance data the linear regression is trained on (resulting in a poor model fit).

Summing up, we have identified the following best parameter values: *WaCkypedia* as a source corpus, with a matrix containing the *20k/50k* most frequent contexts for AP and MITCHELL and the top *100k* contexts for BATTIG and ESSLLI. An *undirected 2-4 word* context window is the best choice for MITCHELL, AP and ESSLLI, while a clear *4-word* window suits BATTIG best. *Cosine* is the best distance metric for all datasets and *neighbor rank* the best relatedness index.

Reduced setting In this section, we discuss the best parameter values for the clustering datasets in the reduced setting.

We start with the known interaction between *feature score* and *feature transformation*, displayed in figure 6.58 to 6.61. The interaction plots show the behavior we are meanwhile familiar with: significance measures (*simple-ll, t-score and z-score*) reach the best performance in combination with *log transformation*: this combination is a robust choice for all datasets. Globally, we observe the shift in patterns from an unreduced to a reduced setting, with almost all measures showing the need of a stronger de-skewing than in the unreduced setting, *log* substituting *root*, which in turn substitutes *no transformation*. Another notable difference with the unreduced setting (and, to a lesser extent, to the other datasets in the SVD runs) is the fact that an untransformed *MI* (corresponding to the PPMI commonly adopted in the literature) is almost equivalent to *simple-ll* and *log*.

We proceed to set the best value for *window size*, by inspecting its interaction with *metric* (figure 6.62), *transformation* (figure 6.63), and *score* (figure 6.64 for AP and 6.65 for BATTIG). The joint inspection of these plots (and of the remaining ones, not shown here for reasons of space) allows us to draw robust conclusions about the three parameters.

For AP, best performance is predicted at a *2 or 4 word window* in combination with

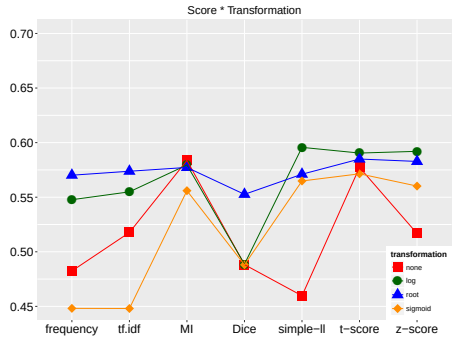

Figure 6.58: AP, red, score / transformation

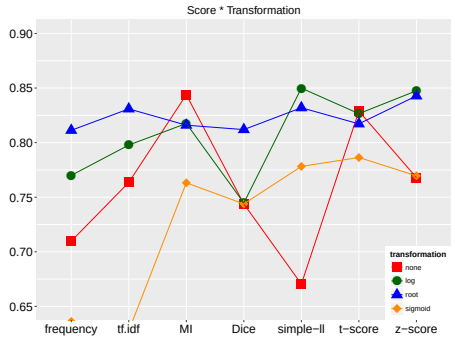

Figure 6.59: BATTIG, red, score / transformation

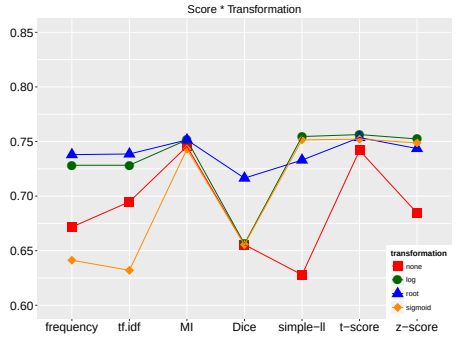

Figure 6.60: ESSLLI, red, score / transformation

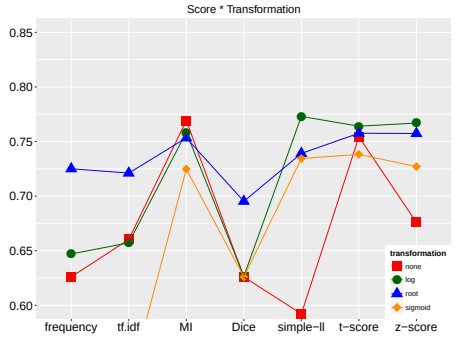

Figure 6.61: MITCHELL, red, score / transformation

cosine distance (figure 6.62), and at a *4 word window* in combination with *log or root* transformation (figure 6.63). The interaction with *feature score* (figure 6.64) contributes significantly to the choice of the best parameter value, as a *4 word window* turns out to be the best value for all involved scores, in particular *simple-ll* (which is also robust at higher dimensionalities). The inspection of the corresponding plots for MITCHELL indicates the same window size. For BATTIG, the interaction with *feature score* (figure 6.65) indicates improvements at a larger window size, *8 words*, and up to 16 without detrimental effects.

Similarly to what we already noticed in the unreduced runs, ESSLLI has a characteristic behavior with respect to the best distance metric, as *manhattan* turns out to be a strong competitor to *cosine*, with conflicting choices as far as other parameters are concerned (see for example the window/metric interaction in figure 6.66). Given their substantial equivalence, the fact that ESSLLI is a very small dataset and thus likely to overfit we choose the "cosine path", which is supported by the general trends we observed in our evaluation. The inspection of the remaining plots makes the choice fall on a *4-word window*. Globally, we observe (a weaker version of) the same shift in the window size pattern which characterized the other tasks: once SVD is applied, all datasets benefit from larger window sizes.

A joint inspection of the interactions involving *corpus*, namely the one with *metric* and the already mentioned *window size* for all datasets, as well as the ones which are dataset specific (see the interaction table in 6.7) indicates in *WaCkypedia* the best corpus for all datasets, confirming to be the best compromise between size and quality.

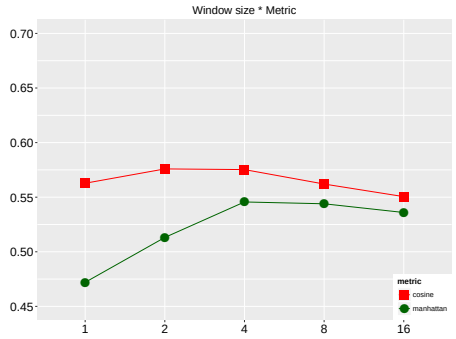

Figure 6.62: AP, red, window / metric

Figure 6.63: AP, red, window / transformation

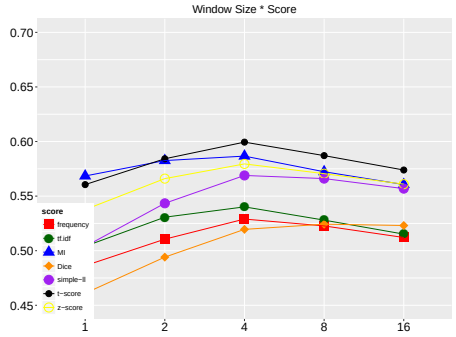

Figure 6.64: AP, red, window / score

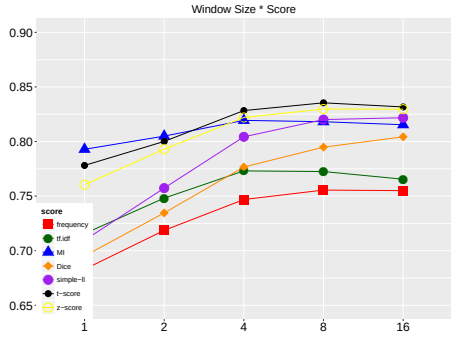

Figure 6.65: BATTIG, red, window / score

The effect plots displaying the interactions involving *number of context dimensions* indicate that in a medium-sized co-occurrence matrix of *50k dimensions* the best robust choice across all datasets, with some dataset-specific variations (see best settings in table 6.9).

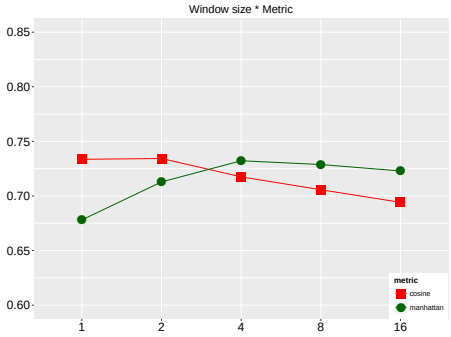

Figure 6.66: ESSLLI, red, window / metric

We now turn to the discussion concerning the best settings for the dimensionality reduction parameters, *number of latent dimensions* and *number of skipped dimensions*, which are involved in a number of interactions, the strongest being the one with *distance metric*. Best performance is predicted with *cosine* and *300 or 500* dimensions; at a full dimensionality, the two distance metrics are equivalent. BATTIG and MITCHELL behave in a comparable way, the latter showing a slight preference for a higher number of dimensions, namely 700. ESSLLI confirms its "special" status with respect to the best choice of metric: *manhattan* is the best choice, with *900* dimensions; with *cosine*, *300* dimensions are sufficient.

A quite clear picture concerning the *number of skipped dimensions* emerges from inspection of its interactions with *metric* and *score* across all datasets. As representative

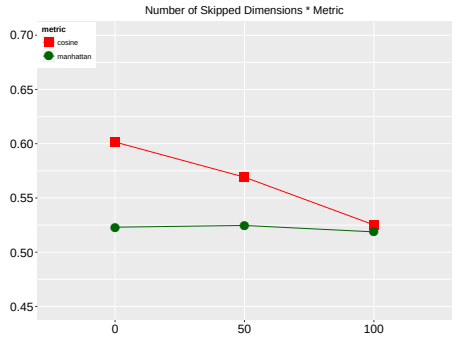

Figure 6.67: AP, red, metric / skipped dim

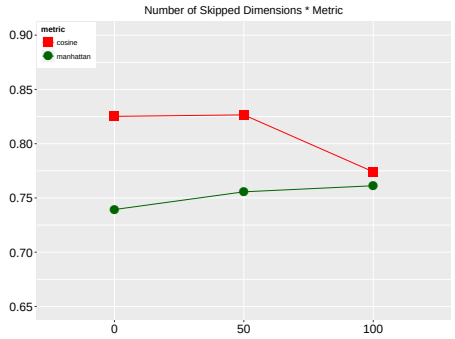

Figure 6.68: BATTIG, red, metric / skipped dim

examples we pick the corresponding plots AP and BATTIG (figure 6.67 and 6.68): they show that skipping dimensions is not necessary to achieve best predicted performance, but skipping the first 50 is not detrimental for BATTIG (and MITCHELL, not shown here). The joint inspection of the effects of the interaction of *number of skipped dimensions* with *score* and *transformation*, respectively, confirm that while for BATTIG and MITCHELL the best choice is not skipping any dimension, skipping the first 50 is, indeed, a viable option in case a robust setting needs to be identified.

Neighbor rank does not participate in any strong interaction, with the exception of its interaction with *metric* for MITCHELL, which identifies in *rank* its best option. As already observed in our discussion of feature ablation, its explanatory power is quite low. Yet, its feature ablation value is above our 0.5 R^2 threshold, and we therefore rely on the the main effect to be sufficient to set its best value to *neighbor rank*. For a more detailed discussion of this effect, see section 6.4.

Best settings In this section, we wrap up our discussion on the best parameters for the clustering datasets by displaying the purity achieved by our best settings. Table 6.8 and 6.9 compare task-specific best settings to a cognitive plausible DSM and to the two PPMI settings. For each dataset, we also report the purity of the best run (see appendix B for the specific settings) as well as the state-of-the-art.[6]

The parameters of the alternative settings are picked as described in section 6.1.1, with a couple of exceptions concerning number of latent dimensions, for which we have clearer indications from the literature, at least for one dataset. Bullinaria & Levy (2012) manipulate SVD on their experiments on the MITCHELL dataset, and while their best result is achieved with many more latent dimensions than in our experiments, from the plots it is comparably clear that *within* our parameter range, the best "literature" setting employs the first 700 dimensions, skipping the first 100. For all other datasets, we employ the neutral setting of 500 latent dimensions, without skipping (Baroni & Lenci, 2010; Levy et al., 2015).

The observations collected in the tables can be summarized as follows:

- Across reduced and unreduced setting, the evaluation methodology allows the identification of robust settings with a reasonably good performance. Just in one case, however, with BATTIG, such best settings come close to the state-of-the art.

[6]State-of-the-art: AP, Rothenhäusler & Schütze (2009); BATTIG, Baroni & Lenci (2010); ESSLLI, Katrenko & Adriaans (2008); MITCHELL, Bullinaria & Levy (2012).

- Differently from what we observed with TOEFL, RG65 and WS353, it is not always the case that the best setting outperforms its competitors. In 4 cases out of 8, the best setting and the best PPMI+ setting are simply two alternative ways to achieve the same performance. It is, however, always the case that the PPMI+ outperforms the "literature" PPMI model, with the exception of the ESSLLI dataset, which has, as discussed, a problematic status given the poor fit of the regression models.

- While SVD brought clear improvements in the other tasks, its positive effect is less marked here: only for AP and ESSLLI the best reduced model outperforms the best unreduced; for BATTIG and MITCHELL, the best reduced and unreduced settings perform at the same level. It is, in any case, recommended to resort to SVD, given the small improvements and the lack of a detrimental effect, and the computational advantages of running the clustering algorithms on lower-dimensional vectors.

- The gap between the best cognitive models and the other models is less marked here than in the other tasks, supporting the impression that, for the clustering task, it has been somewhat easier to identify robust settings alternative to the best ones.

setting	corpus	win	direction	c.dim	exc	score	transf	metric	rel.ind	purity
AP: *best run*: 0.73; *soa*: 0.79										
Best setting	wacky	4	undirected	20000	f	simple-ll	root	cosine	rank	**0.69**
Best cognitive	bnc	8	undirected	100000	f	frequency	log	cosine	rank	0.53
Best PPMI	ukwac	2	undirected	100000	f	MI	none	cosine	dist	0.64
Best PPMI+	wacky	4	undirected	5000	f	MI	none	cosine	rank	0.66
BATTIG: *best run*: 0.99; *soa*: 0.99										
Best setting	wacky	4	undirected	100000	f	simple-ll	root	cosine	rank	0.94
Best cognitive	bnc	1	undirected	100000	f	frequency	log	cosine	rank	0.87
Best PPMI	ukwac	2	undirected	100000	f	MI	none	cosine	dist	0.93
Best PPMI+	wacky	4	undirected	100000	f	MI	none	cosine	rank	**0.99**
ESSLLI: *best run*: 0.93; *soa*: 0.91										
Best setting	wacky	4	undirected	f	100000	z-score	root	cosine	rank	**0.82**
Best cognitive	bnc	2	undirected	f	100000	frequency	log	cosine	rank	0.75
Best PPMI	ukwac	2	undirected	f	100000	MI	none	cosine	dist	0.70
Best PPMI+	wacky	4	undirected	f	50000	MI	none	cosine	rank	**0.82**
MITCHELL: *best run*: 0.97; *soa*: 0.94										
Best setting	wacky	4	undirected	50000	f	simple-ll	root	cosine	rank	**0.86**
Best cognitive	bnc	4	undirected	50000	f	frequency	log	cosine	rank	0.75
Best PPMI	ukwac	2	undirected	100000	f	MI	none	cosine	dist	0.85
Best PPMI+	wacky	4	undirected	5000	f	MI	none	cosine	rank	**0.86**

Table 6.8: Clustering, unreduced – Best settings

setting	corpus	win	direction	c.dim	exc	score	transf	n.dim	dim.skip	metric	rel.ind	purity
AP: *best run*: 0.76; *soa*: 0.79												
Best setting	wacky	4	undirected	20000	f	simple-ll	log	300	0	cosine	rank	0.69
Best cognitive	bnc	4	undirected	20000	f	frequency	log	900	0	cosine	rank	0.61
Best PPMI	ukwac	4	undirected	100000	f	MI	none	900	100	cosine	dist	0.59
Best PPMI+	wacky	4	undirected	20000	f	MI	none	500	0	cosine	rank	**0.70**
BATTIG: *best run*: 0.99; *soa*: 0.99												
Best setting	wacky	8	undirected	50000	f	simple-ll	log	500	0	cosine	rank	**0.98**
Best cognitive	bnc	8	undirected	100000	f	frequency	log	900	0	cosine	rank	0.84
Best PPMI	ukwac	2	undirected	100000	f	MI	none	500	0	cosine	dist	0.96
Best PPMI+	wacky	8	undirected	50000	f	MI	none	500	0	cosine	rank	**0.98**
ESSLLI: *best run*: 0.98; *soa*: 0.91												
Best setting	wacky	4	undirected	100000	f	simple-ll	log	300	0	cosine	rank	0.82
Best cognitive	bnc	4	undirected	50000	f	frequency	log	900	0	cosine	rank	0.80
Best PPMI	ukwac	2	undirected	100000	f	MI	none	500	0	cosine	dist	**0.84**
Best PPMI+	wacky	4	undirected	100000	f	MI	none	300	0	cosine	rank	0.82
MITCHELL: *best run*: 0.97; *soa*: 0.94												
Best setting	wacky	4	undirected	f	50000	simple-ll	log	700	0	cosine	rank	**0.86**
Best cognitive	bnc	4	undirected	f	50000	frequency	log	900	0	cosine	rank	0.66
Best PPMI	ukwac	2	undirected	f	100000	MI	none	700	100	cosine	dist	0.77
Best PPMI+	wacky	4	undirected	f	50000	MI	none	700	0	cosine	rank	**0.86**

Table 6.9: Clustering, reduced – Best settings

6.4 Index of distributional relatedness

As pointed out in the introductory section, the novel contribution of our work is the systematic evaluation the *index of distributional relatedness*, a parameter that has received little attention in DSM research so far, and only in studies limited to a narrow choice of datasets (Hare et al., 2009; Lapesa & Evert, 2013a; Lapesa, Evert, & Schulte im Walde, 2014; Zeller et al., 2014). In this section, we provide a full overview of the impact of this parameter in our experiments, comparing reduced and unreduced runs across all datasets.

Figure 6.69 and 6.70 display the partial effect of *relatedness index* for each dataset, in the unreduced and reduced setting respectively.[7] To allow for a comparison between the different measures of performance, correlation and purity values have been converted to percentages.

As anticipated by the drop in feature ablation from the unreduced to the reduced setting, in all tasks, the advantage of rank over distance is smaller when SVD is applied. A working hypothesis is that sparse spaces are more asymmetrical, leading to a divergence between the predictions of *neighbor rank* and *distance*. Note that the possibility of directly comparing predictions from different spaces is a n inherent advantage of neighbor rank (as soon as rank is calculated over the same vocabulary, of course). We will come back to it in our discussion of the impact of the relatedness index parameter in chapter 7 where the presence of dependency filtered and typed DSMs, notoriously sparser than their window-based counterparts, will provide further support to this interpretation.

Besides the difference between unreduced and reduced runs, the picture emerging from the two plots is quite uniform: *neighbor rank* is almost always the best choice, with the exception of TOEFL where its high computational complexity is clearly not justified; the improvement on the ESSLLI clustering dataset is also fairly small. The degree

[7] For the unreduced runs, we did check all the relevant interactions of relatedness index – and discussed them in the previous sections – to make sure the main effect would not be deceiving. For the reduced runs, the absence of strong interactions makes the inspection of the main effect trustworthy enough.

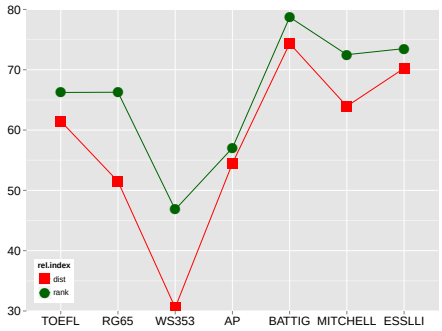

Figure 6.69: Relatedness index: unreduced

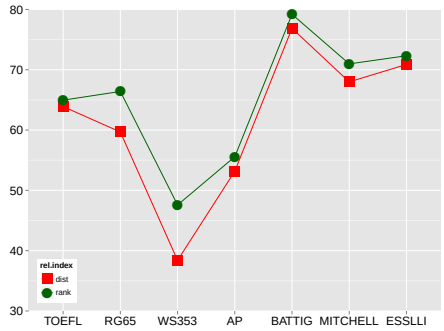

Figure 6.70: Relatedness index: reduced

of improvement over vector distance, however, shows considerable variation between different datasets. The rating task benefits the most from the use of neighbor rank.

While the TOEFL result seems to contradict the substantial improvement of *neighbor rank* found by Lapesa & Evert (2013a) for a multiple-choice task based on stimuli from priming experiments of the type we will be discussing in chapter 8, recall that in a priming setting there are only two choices (consistent and inconsistent prime) rather than four as in TOEFL (correct solution and three distractor). We do not rule out that a more refined use of the rank information (for example, different strategies for rank combinations) may produce better results on the TOEFL.

Moreover, as discussed in section 4.4, we have not yet explored the potential of neighbor rank in modeling directionality effects in semantic similarity. Unlike Lapesa & Evert (2013a), who adopt four different indexes of distributional relatedness (vector distance; forward rank, i.e., rank of the target in the neighbors of the prime; backward rank, i.e, rank of the prime in the neighbors of the target; average of backward and forward rank), we used only a single rank-based index, to keep the number of evaluation runs manageable, making choices which were based on the nature of the tasks at issue.

In the case of the TOEFL experiments, discussed in this chapter, we assumed the multiple choice setting to be inherently directed, and we adopted the rank of the target in the neighbors of the synonym candidates because, in a classification task, taking the rank of the synonym candidates in among the neighbors of the target would have yielded equivalent choices than *distance*. For the multiple choice on the semantic priming datasets, which will be discussed in chapter 8, we employ the rank of the target among the neighbors of the prime, which is the exact replication of the experimental setting. As for ratings and clustering, we decided to go for average rank because it is an unmarked choice in the former case (so we just let rank correct different density of different regions of the semantic space), and the only possible option for the clustering task (where a symmetric measure is expected). We consider the results of this study more than encouraging, and expect further improvements from a full exploration of directionality effects in the rating rating task is concerned. In this connection, we gathered a resource in which speakers judgments have been collected both directions: *word a, word b* vs. *word b, word a* (Lapesa, Schulte im Walde, & Evert, 2014): we consider such resource as a very promising avenue for the exploration of directionality; for more details on this resource we refer the reader to the future work described in section 9.2.

6.5 Best settings

This section wraps up the chapter by addressing the question of whether good general settings can be identified, which perform well across tasks. A side-by-side inspection of the main effects and interaction plots for different data sets allowed us to identify parameter settings which are potentially robust across datasets and even across tasks. Table 6.10 (unreduced runs) and 6.11 (reduced runs) show recommended settings for each task (independent of the particular dataset) as well as a more general setting. In the spirit of the investigations conducted in this chapter, we also report the best PPMI and PPMI+ setting (we remind to the reader that the PPMI+ setting is the best choice of parameters around the MI + no transformation core). Finally, we also identify a robust cognitive setting.

Setting	**Corpus**	**Win**	**Dir**	**C.dim**	**Exc**	**Score**	**Transf**	**Metric**	**Rel.ind**
Best TOEFL	ukwac	2	undir	10k	f	z-score	none	cosine	rank
Best Ratings	wacky	4	undir	20k	f	simple-ll	root	cosine	rank
Best Clustering	wacky	4	undir	50k	f	simple-ll	root	cosine	rank
Best General	wacky	4	undir	50k	f	simple-ll	root	cosine	rank
Best PPMI	ukwac	2	undir	100k	f	MI	none	cosine	dist
Best PPMI+	wacky	4	undir	100k	f	MI	none	cosine	rank
Best Cognitive	bnc	2	undir	100k	f	frequency	log	cosine	rank

Table 6.10: General settings - unreduced

Setting	**Corpus**	**Win**	**Dir**	**C.dim**	**Exc**	**Score**	**Transf**	**Metric**	**N.dim**	**Dim.skip**	**Rel.ind**
Best TOEFL	ukwac	2	undir	5000	f	simple-ll	log	900	100	cosine	dist
Best Ratings	wacky	8	undir	50k	f	simple-ll	log	500	50	cosine	rank
Best Clustering	wacky	4	undir	50k	f	simple-ll	log	500	0	cosine	rank
Best General	wacky	4	undir	50k	f	simple-ll	log	500	50	cosine	rank
Best PPMI	ukwac	2	undir	100k	f	MI	none	500	0	cosine	dist
Best PPMI+	wacky	4	undir	100k	f	MI	none	500	50	cosine	rank
Best Cognitive	bnc	4	undir	100k	f	frequency	log	900	0	cosine	rank

Table 6.11: General settings - reduced

Evaluation results for these settings on each dataset are reported in table 6.12 (unreduced runs) and 6.13 (reduced runs). For RG[65] and WS[353], we report both r and *rho*. For a better comparison, we also report the performance of the best dataset-specific setting (*Best Setting*), as well as the performance of the best run (*Best Run*) and the state-of-the-art in the task (*SoA*).

The performance data displayed in the tables can be summarized as follows:

- In most cases, and in particular in the reduced runs, the general model is close to the performance of the task- and dataset-specific settings. Our robust evaluation methodology has enabled us to find a good trade-off between portability and performance.
- It is often the case that the best PPMI+ models outperform the PPMI ones (for all datasets but TOEFL in the unreduced runs, for TOEFL/RG[65]/WS[353] in the reduced ones, almost always at a dataset-specific level), showing that our methodology can be used to find valid alternative settings.

Dataset	**TOEFL**	**WS**$_r$	**WS**$_{rho}$	**RG**$_r$	**RG**$_{rho}$	**AP**	**BAT**	**ESS**	**MIT**
Best TOEFL	81.25	0.59	0.61	0.75	0.75	0.67	0.79	0.66	0.82
Best Ratings	81.25	**0.67**	**0.69**	**0.88**	**0.88**	**0.69**	0.94	0.77	0.86
Best Clust/Gen	**82.50**	**0.67**	**0.69**	0.87	**0.88**	0.67	0.96	0.80	0.86
Best PPMI	76.25	0.53	0.60	0.67	0.71	0.64	0.93	0.70	0.85
Best PPMI+	72.50	0.64	0.65	0.83	0.83	0.69	**0.98**	0.80	**0.88**
Best Cognitive	68.75	0.36	0.37	0.69	0.68	0.56	0.79	0.75	0.75
Best Setting	81.25	**0.67**	**0.69**	0.82	0.83	**0.69**	**0.98**	**0.82**	0.86
Best Run	87.50	0.73	0.73	0.88	0.88	0.73	0.99	0.93	0.97
SoA	100	0.86	0.83	–	0.81	0.79	0.99	0.91	0.94

Table 6.12: General best Settings - Unreduced

Dataset	**TOEFL**	**WS**$_r$	**WS**$_{rho}$	**RG**$_r$	**RG**$_{rho}$	**AP**	**BAT**	**ESS**	**MIT**
Best TOEFL	**93.75**	0.54	0.62	0.76	0.82	0.61	0.79	0.73	0.73
Best Ratings	86.25	0.69	**0.71**	0.85	0.83	0.63	0.89	0.75	0.76
Best Clustering	75.00	0.65	0.66	0.85	**0.85**	0.67	**0.98**	0.80	**0.88**
Best General	90.00	0.68	0.70	**0.87**	**0.85**	0.67	0.90	0.77	0.83
Best PPMI	75.00	0.57	0.60	0.75	0.77	0.65	0.96	**0.84**	0.87
Best PPMI+	85.00	**0.69**	0.70	0.85	0.82	0.62	0.90	0.80	0.83
Best Cognitive	53.75	0.33	0.33	0.60	0.59	0.60	0.84	0.80	0.86
Best Setting	**93.75**	**0.69**	**0.71**	**0.87**	**0.85**	**0.69**	**0.98**	0.82	0.86
Best Run	98.75	0.73	0.72	0.89	0.86	0.76	0.99	0.98	0.97
SoA	100	0.86	0.83	–	0.81	0.79	0.99	0.91	0.94

Table 6.13: General best Settings - reduced

- General settings are in some cases better than the task-specific ones. This is the case for MITCHELL both unreduced (best PPMI+) and reduced runs, and for RG[65] in the unreduced runs.
- In some cases, however, there is still a discrepancy between the best settings (either dataset-specific, task-specific, or general). One of the potential shortcomings of the methodology proposed in this thesis is the joint interpretation of strong two-way interactions involving the same cluster of parameters: these suggest that looking at three-way interactions could be a better way to go.
- Cognitively inspired models are by far the weakest. The clustering datasets are, however, the ones where the performance of a "general" best cognitive model comes closer to the one of the best setting.

6.6 Summing up

In this chapter, we discussed the results of our evaluation of window-based Distributional Semantic Models, on standard word similarity datasets. Our model selection methodology proved robust to overfitting and capable of capturing crucial parameter interactions. It allowed us to identify parameter configurations that perform well across different datasets within the same task, and, in the majority of the cases, even across different tasks.

We recommend the setting labelled as *Best General*, for both the unreduced (figure 6.12) and reduced (figure 6.13) runs. We believe that many applications of DSMs

(e.g. vector composition) will benefit from using parameter combinations that achieve robust performance in a variety of semantic tasks. While SVD improves performance (although with a gain which varies across datasets), it is worthwhile to identify a best setting for the unreduced runs, as well, because some tasks require count dimensions (e.g., distributional inclusion), and in general it is in many cases important to keep an eye on the properties of models with interpretable dimensions. Moreover, an extensive evaluation based on a robust methodology like the one presented here is the first necessary step for further comparisons of bag-of-words DSMs to different techniques for modeling word meaning, such as neural embeddings (Mikolov, Chen, et al., 2013). Let us now summarize our main findings.

- A cluster of three parameters, namely *score, transformation* and *distance metric*, plays a consistently crucial role in determining DSM performance. These parameters also show a homogeneous behavior across tasks and datasets with respect to best parameter values: *simple-ll, log transformation* and *cosine distance*. These tendencies confirm the results in Polajnar & Clark (2014) and Kiela & Clark (2014). In particular, the finding that sparse association measures (with negative values clamped to zero) achieve the best performance can be connected to the positive impact of context selection highlighted by Polajnar & Clark (2014): ongoing work targets a more specific analysis of their "thinning" effect on distributional vectors.

- *MI + no transformation*, corresponding to the PPMI widely used in distributional semantics, is often a valid option, but not the best one. Given that everybody focusses on frequency or MI, it is no wonder that the potential of transformation has not been uncovered yet: this is a clear contribution of this dissertation.

- Another group of parameters (*corpus, window size, dimensionality reduction parameters*) is also influential in all tasks, but shows more variation with respect to the best parameter values. Except for the TOEFL task, best results are obtained with the *WaCkypedia* corpus, confirming the observation of Sridharan & Murphy (2012) that corpus quality compensates for size to some extent. *Window size* and *dimensionality reduction* show a more task-specific behavior, even though it is possible to find a good compromise in a *4 word window*, a reduced space of *500 dimensions* and *skipping of the first 50* dimensions. The latter result confirms the findings of the clustering experiments by Bullinaria & Levy (2012).

- The *number of context dimensions* turned out to be less crucial. While very high-dimensional spaces usually result in better performance, the increase beyond *20000 or 50000 dimensions* is rarely sufficient to justify the increased processing cost.

- A novel contribution of our work is the systematic evaluation of a parameter that has been given little attention in DSM research so far: the *index of distributional relatedness*. Our results show that, even if the parameter is not among the most influential ones, *neighbor rank* consistently outperforms *distance*: the benefits of using *neighbor rank* clearly outweigh the increased (but manageable) computational complexity. Without SVD dimensionality reduction, the difference is more pronounced.

- In the shift from unreduced to reduced, a more aggressive deskewing of the co-occurrence values is needed: *no transformation* is the best for all but *simple-ll* in the unreduced runs, and drops in performance when SVD is applied; *simple-ll* requires *root* without SVD, *log* with SVD.
- At a very general level, our methodology allows us to identify robust settings across tasks. It can happen, however, that a non-best, yet robust, setting (PPMI+ in the unreduced one) still overperforms our robust best setting.

7

Syntax-based DSMs: Are they worth the effort?

Whereas window-based DSMs adopt a surface-oriented perspective on co-occurrence, dependency-based DSMs adopt a *syntactic* perspective: "nearness" is defined by the presence of a syntactic relation between target and features (e.g. direct object, subject, adjectival modifier). When syntactic relations are used to determine co-occurrence contexts, one speaks of *dependency-filtered* DSMs; if the type of relation is explicitly encoded in the context features (e.g. "subj_dog"), one speaks of *dependency-typed* DSMs. The fortune of syntax-based models in distributional semantics has been mixed. As discussed in chapter 2, early work indicated that syntax-based semantic representations are indeed superior: these studies, however, were restricted to a specific corpus (BNC in Padó & Lapata (2007)) or task (noun clustering in Rothenhäusler & Schütze (2009)), or based on a very specific notion of co-occurrence (Baroni & Lenci, 2010). Meanwhile, extensive evaluation studies and parameter tuning led to significant improvements in the performance of window-based models (Bullinaria & Levy, 2007, 2012). Among recent comparative evaluation studies, only Kiela & Clark (2014) attempt a direct comparison between the parameter spaces of window-based and syntax-based DSMs: window-based models are found to perform better (with the exception of models built from the large Google Books N-gram corpus), but the scope of this comparison is rather limited.

The evaluation presented in chapter 6 has investigated in detail the properties of the window-based parameters, and indicated clear directions for the improvement of their performances (e.g., neighbor rank as an index of distributional relatedness; selection of SVD dimensions). The aim of this chapter is to establish a fair ground for the comparison between window-based and syntax-based DSMs to properly address the question of whether dependency-based models can significantly improve DSM performance if the parameters are properly set, and even in that case, whether the degree of the improvement justifies the increased complexity of the extraction process.[1] In either case, a more thorough understanding of the parameter space will be beneficial for applications that prefer dependency-based DSMs on general grounds, e.g. because of an integration with syntactic structure (Erk et al., 2010). While the evaluation reported in this chapter does not encompass predict-type models, we believe that our findings also apply to the usefulness of dependency information in neural word embeddings (Levy & Goldberg, 2014a).

We take as a reference point the large parameter set evaluated for window-based models in chapter 6, and carry out a parallel evaluation for dependency-based DSMs

[1] Part of the material presented in this chapter has been published in Lapesa & Evert (2017).

using the same tasks, datasets and parameters. In addition, we introduce some parameters which are specific to syntax-based models, such as the parser used and the type of allowed dependency relations. Similarly to what we have done for the window-based experiments, we do not discuss the distribution of performances here (the reader can find them in appendix C) and, instead, focus on the interpretation of the output of our linear regression methodology. The fit of the regression models, displayed in table 7.1, is good for all datasets and displays the same trends we have already observed for the window-based models: within evaluation task, smaller datasets have a lower fit than the larger ones (ESSLLI, in particular, confirms the window-based tendency to be the noisiest of the entire evaluation setting); for most cases, the fit of the reduced runs is lower than the one of the unreduced runs, as the dimensionality reduction parameter potentially introduces additional noise.

Dataset	Filtered		Typed	
	Unreduced	Reduced	Unreduced	Reduced
TOEFL	88.1	88.3	88.0	89.5
RG65	92.2	87.0	92.4	86.8
WS353	94.2	88.2	93.5	90.7
AP	87.0	82.9	87.5	88.2
BATTIG	80.2	74.2	79.4	78.8
ESSLLI	70.4	56.9	66.9	67.5
MITCHELL	84.9	74.3	82.9	77.4

Table 7.1: Dependency models: adjusted R^2 across settings and datasets

Given the large number of parameters to explore and the need for a qualitative interpretation of the results, in this chapter we adopt a different narrative with respect to chapter 6. We keep the focus of our discussion on three datasets: TOEFL (for comparison with the literature), WS353 (the largest ratings dataset in our experimental set, and the one with the highest R^2 values), and AP (the largest clustering dataset, and the one with the highest R^2 values).

Throughout the chapter, we discuss the main trends in the performance of our dependency-based DSMs, highlighting the differences from window-based results, and those among different dependency-based settings (filtered vs. typed, reduced vs. unreduced).

7.1 Feature ablation

Figures 7.1 and 7.2 visualize the feature ablation values of all evaluated parameters in the dependency-filtered and dependency-typed setting, for reduced and unreduced runs. Tables 7.2 to 7.5 display all major interactions (partial $R^2 > 0.5$) in the four settings.

Figure 7.1: Dependency filtered: Feature ablation. Left: unreduced; Right: reduced.

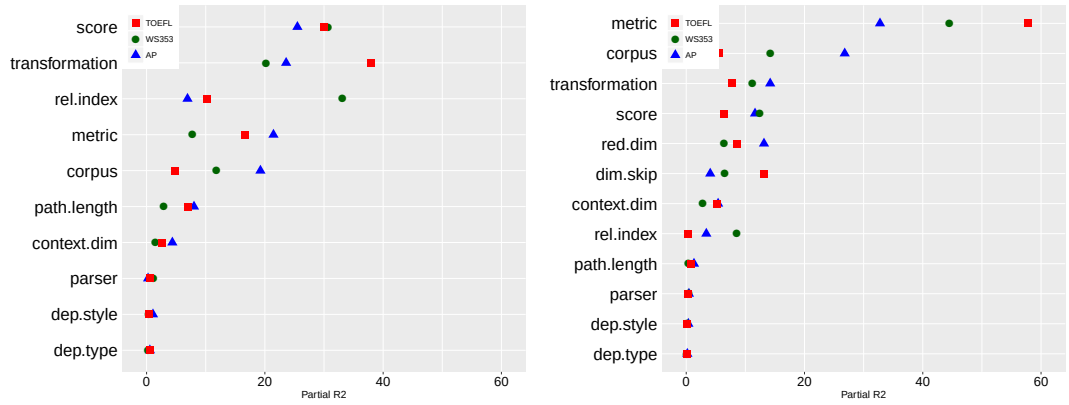

Figure 7.2: Dependency typed: Feature ablation. Left: unreduced; Right: reduced.

Let us start by summarizing the observations concerning the feature ablation trends, in comparison to what we observed for the window-based models:

- In all four settings, we observe a strong effect of score and transformation, which dominate the feature ablation ranking in the unreduced runs. Interestingly, the relative position of the two parameters flips in the typed/reduced setting as compared to the other three settings. In practice, this means that while in the unreduced setting the regression analysis identifies a strong variation among the different scores, such difference is neutralized in SVD. This could be interpreted either as an effect of SVD (which improves the representation no matter which the underlying score is) or as a negative effect of the high sparsity of the space which is input to SVD (the space being too sparse for SVD to be able to boost performance).

- Score and transformation are outscored by metric in the reduced setting, due to the strong interactions metric entertains with almost all other parameters. A gain in ablation power for metric in the reduced setting is something we have already observed in the window-based experiments, in particular for TOEFL: indeed, this is the case in the dependency-based experiments as well, with TOEFL (red square in the plots) being the dataset for which the effect of metric is the strongest.

	TOEFL	WS	AP
score × transf	9.5	5.8	7.3
transf × metric	1.0	0.5	5.1
score × rel.index	–	4.8	1.5
score × metric	0.8	0.5	3.5
corpus × score	0.9	1.1	1.8
score × cont.dim	1.0	–	1.4
corpus × metric	0.5	–	1.1
dep.group × path.length	0.9	–	–
metric × cont.dim	0.7	–	–
path.length × transf	–	–	0.6
path.length × score	0.6	–	–
corpus × rel.index	0.5	–	–

Table 7.2: Filt, unred: interactions

	TOEFL	WS	AP
score × transf	8.3	11.2	8.6
metric × dim.skip	4.0	1.1	3.4
metric × red.dim	0.6	1.4	3.6
score × metric	1.3	1.5	1.8
metric × cont.dim	1.0	1.2	0.6
corpus × metric	–	0.9	1.9
transf × dim.skip	0.6	0.8	0.5
score × cont.dim	0.7	0.8	0.8
corpus × score	0.6	1.0	–
score × dim.skip	–	0.8	–
path.length × transf	–	–	0.8
transf × metric	–	0.7	–
score × rel.index	–	0.6	–
path.length × metric	–	–	0.6
metric × rel.index	–	–	0.5

Table 7.3: Filt, red: interactions

	TOEFL	WS	AP
score × transf	10.4	7.9	6.5
transf × metric	2.1	1.0	5.9
corpus × score	1.8	2.0	1.6
score × metric	0.7	0.5	3.6
metric × rel.index	1.7	–	0.8
score × rel.index	–	1.5	0.6
score × cont.dim	1.2	0.6	1.0
corpus × transf	1.2	–	–
metric × cont.dim	0.7		–
transf × rel.index	–	0.5	–

Table 7.4: Typed, unred: interactions

	TOEFL	WS	AP
score × transf	2.4	5.0	5.7
metric × dim.skip	4.9	2.2	1.2
metric × cont.dim	3.3	2.0	2.3
metric × red.dim	–	1.3	4.7
corpus × metric	–	1.0	4.6
corpus × score	–	1.0	0.6
transf × metric	0.9	–	–
score × metric	0.8	–	–
transf × dim.skip	–	0.6	–
corpus × transf	–	–	0.5
metric × rel.index	–	–	0.5
cont.dim × dim.skip	–	–	0.5

Table 7.5: Typed, red: interactions

- Relatedness index is the third most powerful parameter in the unreduced settings, for both filtered and typed models – turning out to be slightly more powerful in the (sparser) dependency setting than in the window-based one. Similarly to what we have already observed in the window-based experiments, this parameter loses power from the reduced to the unreduced setting.

- Source corpus is more powerful here as compared to the window-based setting. Overall, its explanatory power follows the rank reduced-typed > reduced-filtered > unreduced-typed/filtered. By looking at the best parameter values we will clarify whether this an effect of corpus size (larger corpora providing better coverage in an intrinsically sparse setting) or parsing quality (sentences from smaller corpora being shorter hence easier to parse).

- Path length is just a middle-range predictor of model performance – for WS, it is on the weak side of the ranking for all four settings. Overall, path length is weaker in a dependency-typed setting than in a filtered setting (no significant improvements in performance have to be expected going from single-step paths to multiple-step paths): this is probably due to the fact that one-step paths (which have been commonly employed in the literature on syntax-based DSMs) constitute a robust

"core" of the space. Path length loses power from an unreduced to a reduced setting: note that the corresponding parameter in the window-based experiments (window size) exhibited the opposite trend, gaining power for TOEFL and AP and staying stable for WS. Overall, this can be interpreted as an effect of the fact that the information introduced by including longer paths is too sparse and it is neutralized as noise by dimensionality reduction.

- SVD-related parameters impact DSM performance to the same extent in filtered and typed setting, occupying the middle range of the ablation ranking.

- As for the number of context dimensions, the feature ablation power occupies a lower rank, just above the dependency-specific parameters.

Dependency-specific parameters (dependency style, dependency group, and parser) have an extremely weak explanatory power. The only effect above our 0.5 R^2 threshold is the interaction between dependency group and path length, for TOEFL and in the unreduced/filtered setting. The take-home message for this result is that the manipulation of dependency-specific parameters is not bound to affect DSM performance significantly. Such parameters can be set to default values that are identified based on computation time (Malt parser being quicker than Stanford), degree of lexicalization (dependency style: basic dependencies being less lexicalised – and more assumption-free – than the CCprocessed ones), and complexity of the syntactic relations involved (dependency group: core relations being part of the immediate argument structure of the target).

7.2 Dependency filtered models

In this section, we discuss the results of the linear regression analysis on the performance of the dependency-filtered DSMs, for both unreduced and reduced runs, keeping the focus on the comparison to window-based models and relying on the same interpretative criteria as in the previous chapter.

Recall that dependency-filtered DSMs can be seen as a special case of bag-of-words DSMs in which context selection operates on the basis of the syntactic information encoded in the dependency graphs. Before getting deeper into the discussion of the specific effects, let us elaborate on the interpretation of the evaluated parameters in a dependency-filtered setting:

- The window size parameter from the window-based analysis maps (conceptually and practically) onto path length.

- Dependency group operates as a context selection criterion on the edge label, allowing into the co-occurrence matrix only nodes which are connected to the target by a core dependency relation. It has no direct correspondent in the window-based experiments, as there we did not conduct experiments on the edge labels to select contexts.

- In the context of a dependency-filtered model, the difference between a basic and collapsed/CCprocessed dependency style is in terms of path length. For example, the prepositional objects are two steps away from their head nodes in the basic

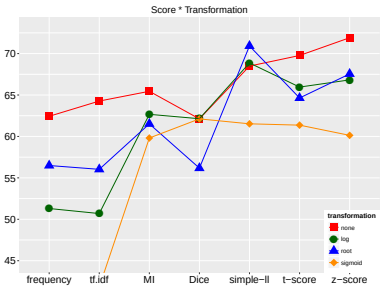

Figure 7.3: TOEFL-filt/unr

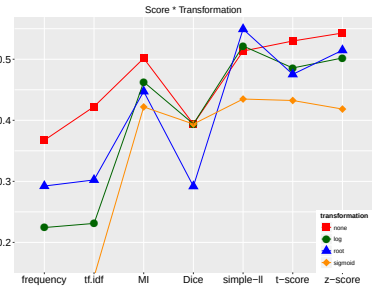

Figure 7.4: WS-filt/unr

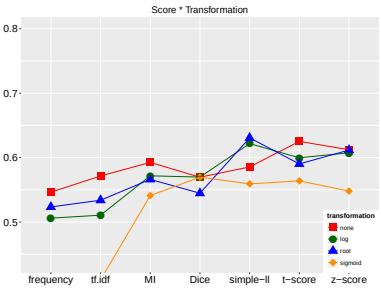

Figure 7.5: AP-filt/unr

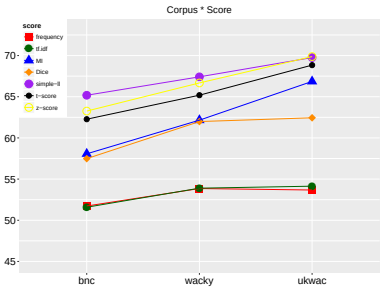

Figure 7.6: TOEFL-filt/unr

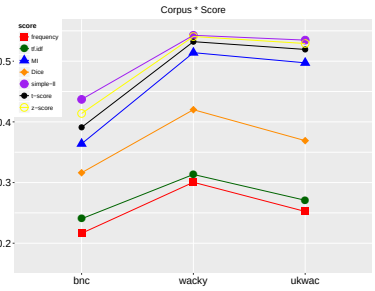

Figure 7.7: WS-filt/unr

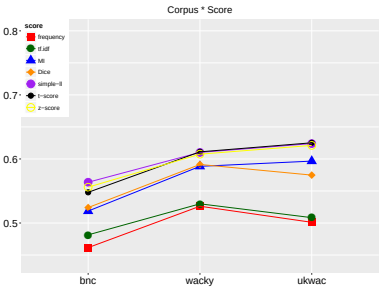

Figure 7.8: AP-filt/unr

dependency style (via the preposition node), while they are just one step away in the collapsed/CCprocessed graph (where a direct path is established, whose edge is labelled with the preposition).

Unreduced Runs Let us start our discussion of the effects for dependency-filtered DSMs, following the ranking of interactions in table 7.2.

In what represents a signature result of the experiments presented in this thesis, we find *score* and *transformation* at the top of the feature ablation ranking, for all datasets. We display the interaction plots in figures 7.3 (TOEFL), 7.4 (WS), and 7.5 (AP). They show that a robust parameter combination, namely *simple-ll* with a *root* transformation, can be identified for all three datasets; as an alternative, *z-score* without transformation is also a valid combination – and the best one for TOEFL. Not only does the strong ablation value of *score* and *transformation* generalize from the window-based to the dependency-based experiments, but also their best parameter values are the same.

The interactions involving *metric*, not shown here, confirm that *cosine* is the robust best choice for all datasets and so is *neighbor rank* as an index of distributional relatedness (see section 7.4 for a detailed discussion of the effects of relatedness index).

The interaction between *corpus* and *score*, displayed in figures 7.6 (TOEFL), 7.7 (WS), and 7.8 (AP), identifies in *UkWaC* the best corpus in combination with the strongest scores (*simple-ll* and *z-score*), with the exception of WS, for which WaCkypedia and UkWaC are equivalent in terms of predicted performance. This is the first difference we observe with respect to the window-based experiments: when dependencies are employed to select relevant contexts, a larger corpus is needed. This is likely to be an issue of coverage/quality of the representations: more occurrences of the target

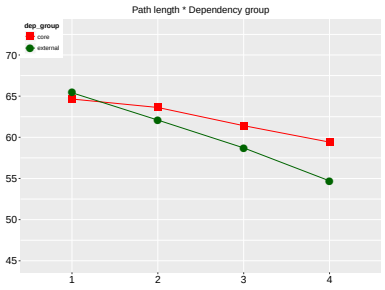

Figure 7.9: TOEFL-filt/unr

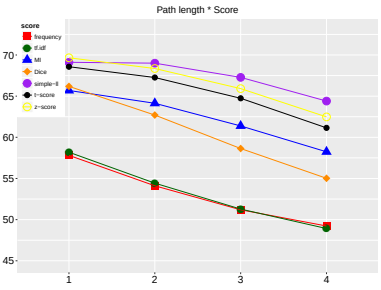

Figure 7.10: TOEFL-filt/unr

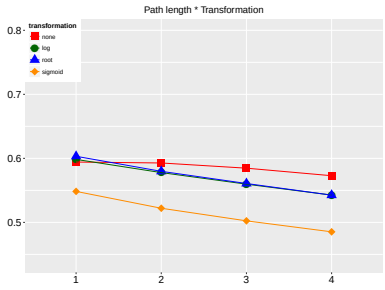

Figure 7.11: AP-filt/unr

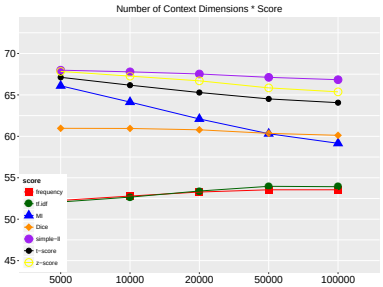

Figure 7.12: TOEFL-filt/unr

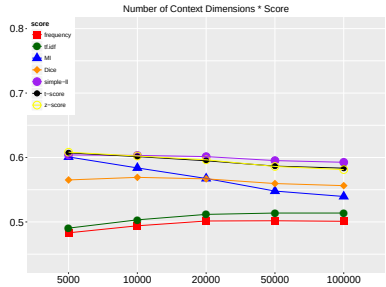

Figure 7.13: AP-filt/unr

words are necessary to produce the best representations.

Let us now look into the interaction between score and number of context dimensions, displayed in figure 7.6 (TOEFL) and 7.8 (AP). Similarly to what we have observed in the window-based runs, TOEFL shows a preference for fewer contexts and, in general, medium-sized co-occurrence matrices are sufficient for best results. Yet, with respect to the window-based runs, the dependency filtered models show a slight preference for the most frequent contexts (*5k* or *10k*) or, at least, no clear advantage is predicted for the use of larger co-occurrence matrices (which involve lower-frequency contexts). WS does not present any strong interaction with number of context dimensions: an inspection of the simple effect (cf. supplementary material) confirms the general tendency towards higher-frequency contexts. To round up the interpretation of the behavior of this parameter, we need to take into account the different dynamics that bring a context word into the window-based vs. dependency-filtered matrix: and to do that, we need to bring path length into the picture.

As anticipated in the feature ablation section, *path length* has an intermediate to weak effect on model performance. The simple effect (cf. supplementary material) shows that a path of length 1 is the best choice for all datasets – longer paths are detrimental for TOEFL and AP, and irrelevant for WS (where the parameter has a minimal impact anyway). *Path length* only participates in three weak interactions (below 1% R^2): for TOEFL, it interacts with *dependency group* (figure 7.9) and *score* (figure 7.10). The interaction with *dependency group* indicates that, while adding the *external* dependencies to the *core* ones produces a minimal improvement at the shortest path, it is clearly detrimental at longer paths. Once more, our results point in the direction of simplicity. The interaction between *score* and *path length* identifies in *simple-ll* the only measure that is robust at longer paths – this tendency is comparable to what we have already observed for the interaction between *score* and *window size* in the window-based models

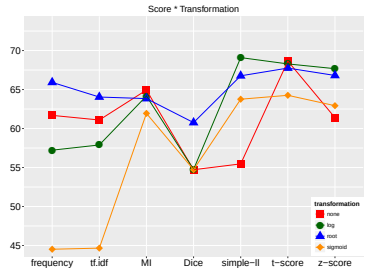

Figure 7.14: TOEFL-filt/red

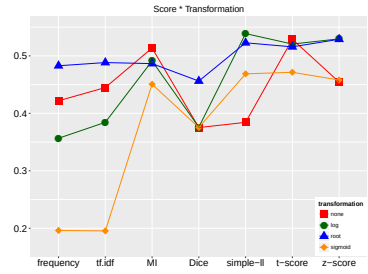

Figure 7.15: WS-filt/red

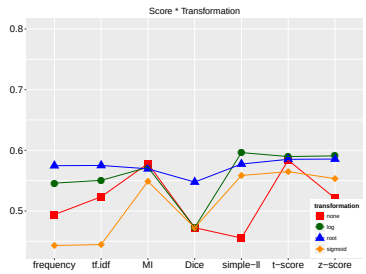

Figure 7.16: AP-filt/red

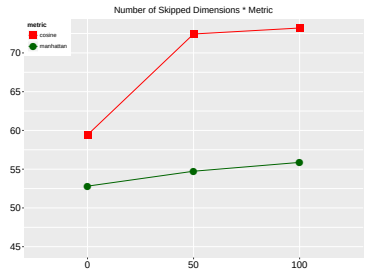

Figure 7.17: TOEFL-filt/red

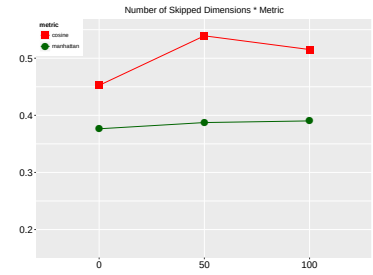

Figure 7.18: WS-filt/red

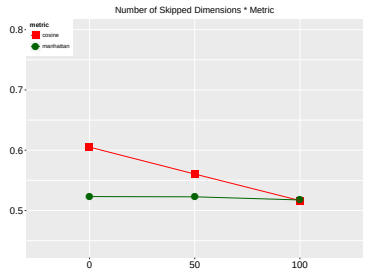

Figure 7.19: AP-filt/red

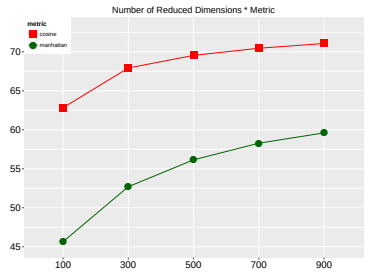

Figure 7.20: TOEFL-filt/red

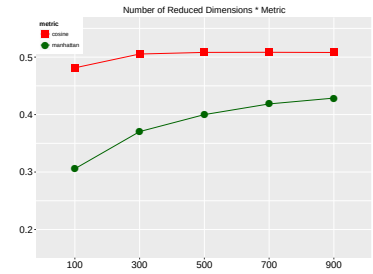

Figure 7.21: WS-filt/red

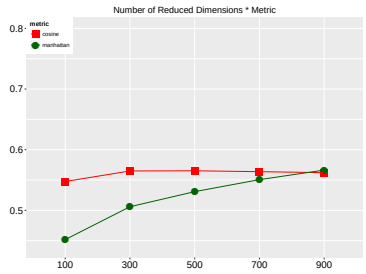

Figure 7.22: AP-filt/red

(figure 6.11).

For AP, the interaction between *path length* and *feature transformation* (figure 7.11) confirms that paths of length 1 are still the best choice – however, without any transformation, paths of length 2 turn out to be equivalent to those of length 1, and, in general, even a length of 3 is not as detrimental as it is in combination with the other transformations. The fact that the gain with log and root at path length 1 is minimal, that there is a drop in performance at longer paths, and that performance is relatively stable at longer paths without any transformation, has to do with the aggressive deskewing: the take-home message is that in a very sparse space it is good to preserve the difference between high values and (extremely) low values.

Reduced Runs Let us now turn to the discussion of the best parameter settings for the reduced runs, following the ranking identified in the feature ablation (figure 7.1)

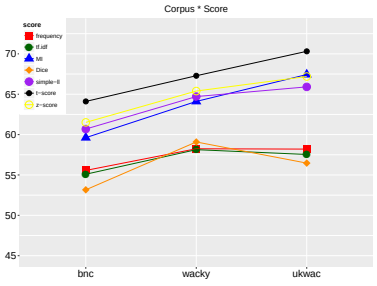

Figure 7.23: TOEFL-filt/red

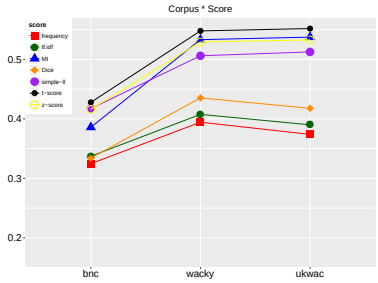

Figure 7.24: WS-filt/red

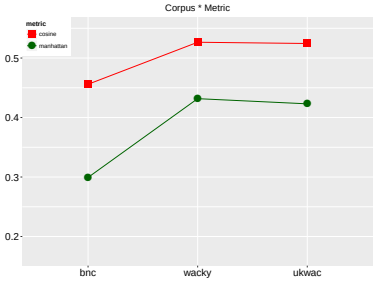

Figure 7.25: WS-filt/red

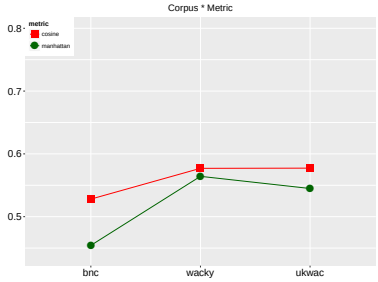

Figure 7.26: AP-filt/red

and in the corresponding interaction table (figure 7.3).

The strong interaction between *score* and *transformation*, displayed in figures 7.14 to 7.16 for the three datasets indicates a preference for simple log-likelihood with log transformation or MI without any transformation. Comparing reduced to unreduced, we observe the pattern of shift in best parameters already observed for the window-based models: *log* takes over the role of *root* as best transformation, and *no-transformation* is not anymore as robust across parameters as it used to be in the unreduced runs.

Manhattan *metric* always performs much worse than cosine distance; the different behavior of the two metrics also accounts for most of the interactions listed in table 7.3. Its interaction with the SVD parameters are displayed in figures 7.17, 7.18, and 7.19 (number of skipped dimensions) and 7.20, 7.21, and 7.22. Once again, we observe similar trends as in the window-based models: need for skipping 50/100 dimensions for TOEFL, 50 for WS, none for AP, and a decreasing number of necessary reduced dimensions, namely 900 for TOEFL, 500 for WS, 300 for AP.

A joint inspection of the interactions involving *source corpus* (figures 7.23 to 7.26) shows that two bigger *corpora* are always a better choice, with a clear preference for *ukWaC* in the case of TOEFL. *Neighbor rank* outperforms distance, but the increased computational cost may only be justified for AP and WS – a more detailed discussion of this effect is provided in section 7.4.

The interactions involving *context dimensions* (cf. supplementary material), indicate that while the top *5k* are sufficient for TOEFL, more dimensions (20k or 50k) are necessary for the other two datasets. As far as *path length* is concerned, relevant interactions only involve AP – their inspection together with the main effect (cf. supplementary material), indicates that the preference for the shortest paths identified for the unreduced runs still holds for TOEFL and AP, while the co-occurrence information

coming from longer paths becomes beneficial for WS, as soon as SVD is involved.

7.3 Dependency typed models

In this section, we discuss best parameter settings for the experiments involving the dependency typed DSMs. Recall that:

- Dependency-typed DSMs exploit the information contained in the dependency edges to build finer-grained features (e.g., *is subject of the verb bark*): in this perspective, dependency-typed DSMs stand to dependency-filtered DSMs in a comparable relation to the one in which window-based DSMs enconding relative position with respect to the target (e.g., *it occurs to the left of the verb bark*) stand to the position-unaware ones (e.g., *it occurs in the context of the verb bark*). Needless to say, dependency-typed features also produce much sparser spaces.

- Within the dependency-typed DSMs, the *dependency-style* parameter regulates the degree of lexicalization of the links (e.g., *is a prepositional object of the verb bark* vs. *is a prepositional object of the verb cut, and the head preposition is with*). Once again, the more context features are lexicalized, the sparser is the space.

Unreduced Runs Let us now discuss the best parameter settings for the unreduced runs. Recall that this set of experiments corresponds to the sparsest setting among those presented in this dissertation.

We start from the familiar *score* and *transformation* interaction, illustrated in figure 7.27 for TOEFL, 7.28 for WS, and 7.29 for AP. While the best parameter combinations are quite similar to those of the corresponding dependency filtered runs (for TOEFL and AP the best combination is *z-score* and *no transformation*, for WS we observe a slight preference for *simple-ll* with a *root* transformation), what we observe here is that performances are overall lower and there is less variation across the different combinations (in particular for AP and WS). We observe a notable drop in predicted performance for the option *MI+no transformation*, in particular for TOEFL and, to a lesser extent, WS; this is not surpising given the known tendency of MI to overestimate low-frequency items (which are likely to result from the use of lexicalised context features) and it is a consistent pattern also with respect to the window-based experiments.

Figures 7.30, 7.31, and 7.32 illustrate the interaction between feature *score* and *source corpus* for the three datasets. *WaCkypedia* is the most robust choice across all corpora. Note that the interaction plot for *MI* shows a strong preference for *UkWac* – together with the performance drop of *MI* observed for TOEFL, this result supports our interpretation that in this setting *MI* is negatively affected by low-frequency features (hence it has better performances with the largest corpus).

The plots displaying the *number of context dimensions* (figures 7.33, 7.34, and 7.35) confirm the property of *simple-ll* to benefit from (or at least not be negatively affected by) the presence of low-frequency contexts: as a matter of fact, *100k* dimensions is the best parameter value for AP and not detrimental for TOEFL and WS. On the other hand, we notice for *MI* the known performance drop when low frequency contexts get into the picture.

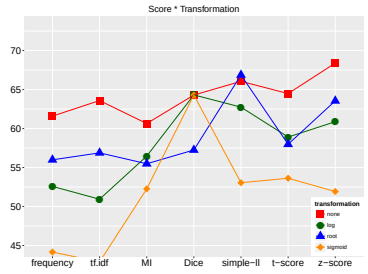

Figure 7.27: TOEFL-typ/unr

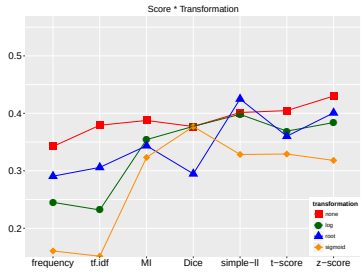

Figure 7.28: WS-typ/unr

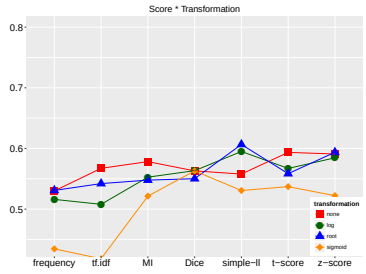

Figure 7.29: AP-typ/unr

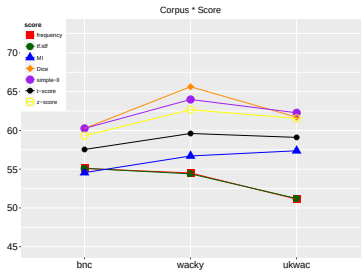

Figure 7.30: TOEFL-typ/unr

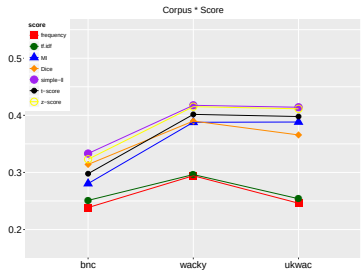

Figure 7.31: WS-typ/unr

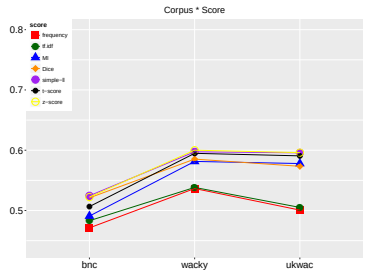

Figure 7.32: AP-typ/unr

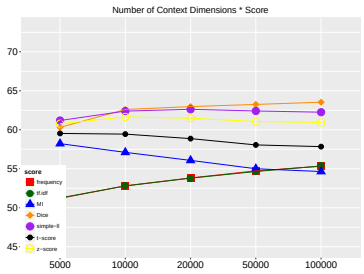

Figure 7.33: TOEFL-typ/unr

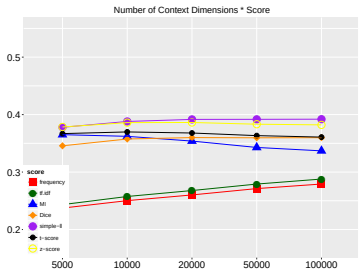

Figure 7.34: WS-typ/unr

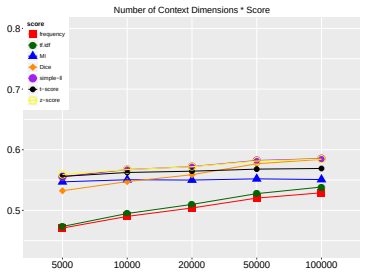

Figure 7.35: AP-typ/unr

Other effect plots are not shown here, but are reported in the supplementary material. Their inspection allows to set the *distance metric* to *cosine*, *relatedness index* to *neighbor rank*, and *path length* to *1* for all datasets.

Reduced Runs In this section we discuss the best parameter settings of the experiments involving SVD.

As shown in the feature ablation plot and in the corresponding interaction table, metric is the strongest parameter – and it heavily interacts with many of the other involved parameters, as we have already observed for the other experimental settings. The inspection of the corresponding interaction plots reveals no surprises:

- *cosine* is the best *distance metric* in all involved combinations – and the difference between the two metrics is more acute in this setting (e.g., for AP *manhattan* does not ever come close to *cosine*, as it had happened in a handful of cases with window- or dependency-filtered DSMs).

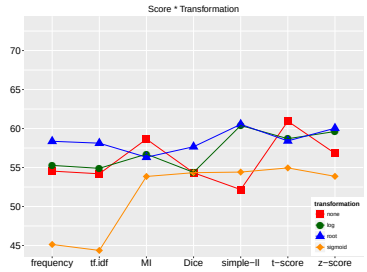

Figure 7.36: TOEFL-typ/red

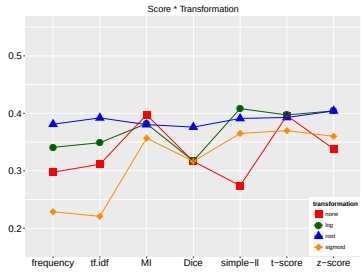

Figure 7.37: WS-typ/red

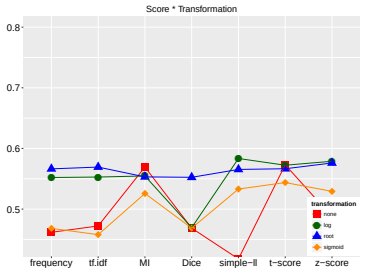

Figure 7.38: AP-typ/red

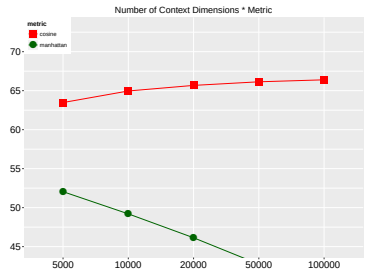

Figure 7.39: TOEFL-typ/red

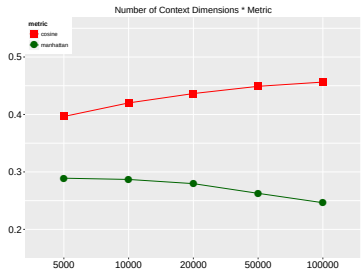

Figure 7.40: WS-typ/red

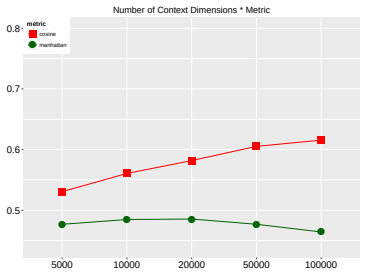

Figure 7.41: AP-typ/red

- The *SVD parameters* interact with metric, and they also show the common trends: it is beneficial to skip the first 50/100 dimensions for TOEFL, the first 50 for WS, while no skipping is the best choice for AS (though the differences between the different values are less sharp in the case of the clustering dataset); as far as reduced dimensions are concerned, 300/500 are sufficient but more are not detrimental.

- As for the interactions of *metric* with *corpus*, we observe the known preference for the two larger corpora, with *WaCky* already robust enough across datasets.

The interaction between *feature score* and *transformation* is still the strongest overall (cf. table 7.5), but it has lost explanatory power (it is even outscored by the interaction between metric and skipped dimensions in the case of TOEFL). This is clear from the interaction plots, displayed in figures 7.36, 7.37, and 7.38. There is strikingly less difference between the performances of different parameter combinations: as a matter of fact performances are "squished" in the lower bands of the distribution – in other words, no combination performs really well. The unreduced/reduced shift in terms of best transformation we identified in both window-based and dependency-filtered models (*no transformation* being overscored by *root*, which in turn would be overscored by *log*) is also not so prominent here. Once again, the interpretation of this fact is related to the high sparsity of the underlying space: the close tie between *root* and *log* in combination with *simple-ll* shows that the association measure has produced less skewed values and, despite its robustness to low-frequencies, in this case it seems to not have assigned very high scores to salient features.

The inspection of the interactions involving *number of context dimensions* (figures 7.39, 7.40, and 7.41) reveals what is a clear tendency of the dependency-typed models

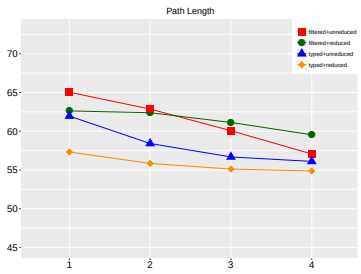

Figure 7.42: TOEFL

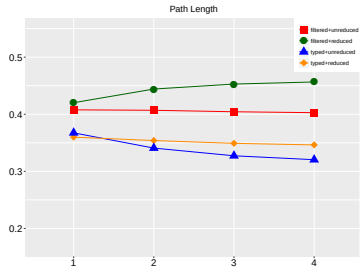

Figure 7.43: WS

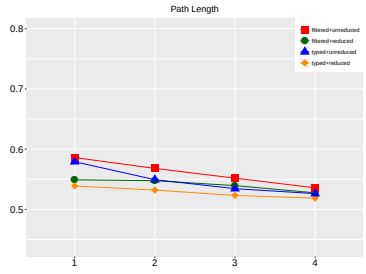

Figure 7.44: AP

with respect to the dependency-filtered ones: the more context dimensions, the better the performance. The message here is that, when the features are so granular, even the lowest frequency ones can contribute significantly to a better characterization of the target words in the datasets – very small context matrices risk to underrepresent certain target words.

Let us conclude with the effect of *path length*, which we have already observed to be quite weak in this setting. Given the absence of (strong) interactions we can resort to the main effect plot, and in doing so we also use the opportunity to compare the main effect across the different settings (filtered vs. typed, reduced vs. unreduced), in the plots shown in figures 7.42 (TOEFL), 7.43 (WS), and 7.44 (AP). The plots show that within the general tendency of dependency-based models to reach best performances at *shortest paths*, the detrimental effect of longer paths can be rescued by dimensionality reduction only in the case of dependency-filtered models. Even for WS, which shows a clear improvement at longer paths in the filtered/unreduced setting, the best value for a dependency-typed model is still 1: heavy lexicalization makes features based on longer paths just too fine-grained – even resorting to a larger corpus like UkWaC does not help in this case.

7.4 Index of distributional relatedness

In this section, we discuss the effect of relatedness index across all the dependency-based settings. Recall that in the feature ablation study we have already observed that relatedness index loses power from unreduced to reduced setting – indeed, the effect plots in figures 7.45, 7.46 and 7.47 for the four combinations of filtered vs. typed, reduced vs. unreduced reflect the loss of predictive power in that the two parameter values, distance and neighbor rank, get closer. This finding is in line with the interpretation we proposed in section 6.4 for the window-based models, namely that the sparser the space is, the more asymmetric it is.

The plots have a clear take-home message: *neighbor rank* is always the best performing value. From the interaction tables we know that in the unreduced settings it also participates in interactions with other parameters: for the filtered models, it interacts with feature score (AP, WS); for the typed models, it interacts with score (WS) and metric (TOEFL, AP). The corresponding plots are shown in the supplement, but it should be sufficient to point out here that none of them contradicts the main trend: in other words, there is no case in which in combinations with certain scores *distance* outperforms *rank* – it is just that the extent to which *rank* outscores *distance* varies

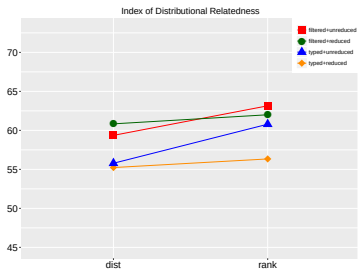

Figure 7.45: TOEFL

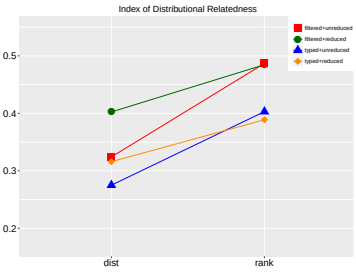

Figure 7.46: WS

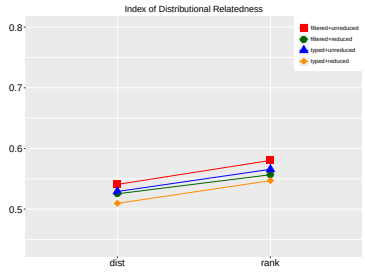

Figure 7.47: AP

across scores (which is slightly less marked for association measures and manhattan distance).

7.5 Best settings

Tables 7.6 and 7.7 report the robustly optimal parameter settings for dependency-filtered and dependency-typed models. For each dataset, we list the parameter settings identified by inspecting the effect plots and specify their performance (*b.synt*). For comparison, we also report the performance of the optimized window-based DSM from chapter 6 (*b.win*), and of the state of the art for the task (*soa*).

	corpus	parser	d.gr	d.st	p.len	c.dim	score	transf	metric	r.ind	b.synt	b.win	soa
Dependency filtered													
TOEFL	ukwac	malt	core	basic	1	5k	z-score	none	cosine	rank	72.5	93.7	100
WS	wacky	malt	core	basic	1	5k	simple-ll	root	cosine	rank	0.67	0.69	0.83
AP	ukwac	malt	core	basic	1	5k	simple-ll	root	cosine	rank	0.54	0.69	0.79
Dependency typed													
	corpus	**parser**	**d.gr**	**d.st**	**p.len**	**c.dim**	**score**	**transf**	**metric**	**r.ind**	**b.synt**	**b.win**	**soa**
TOEFL	wacky	malt	core	basic	1	100k	z-score	none	cosine	rank	80.0	93.7	100
WS	wacky	malt	core	basic	1	50k	simple-ll	root	cosine	rank	0.53	0.69	0.83
AP	wacky	malt	core	basic	1	100k	z-score	none	cosine	rank	0.68	0.69	0.79

Table 7.6: Unreduced runs - best settings

Our methodology allowed us to identify robust parameter settings for each task. However, only in one case such best settings manage to beat the best window-based model identified in chapter 6: in the dependency-based, reduced setting and for the noun clustering task.

Table 7.8 reports the parameter values of *general* settings identified by comparing the effect plots to find values robust across all tasks. Their performance on TOEFL, WS, and AP as well as on the remaining datasets from the window based evaluation is reported in table 7.8. The table also reports the performance of the *reference* setting for each task (WS for RG; AP for ESSLLI, MITCHELL, BATTIG) on the remaining datasets. Such reference setting is trained on the largest – hence more reliable – dataset per task which can be regarded as a development set employed to tune parameters that can then be tested on the smaller datasets. The best performance per dataset is highlighted in bold.

The results listed in table 7.9 confirm the tendency of all clustering datasets to achieve their best results in a dependency-typed (and SVD-reduced) setting.

	Dependency filtered														
	corpus	**parser**	**d.gr**	**d.st**	**p.len**	**c.dim**	**score**	**transf**	**metric**	**d.sk**	**r.dim**	**r.ind**	**b.synt**	**b.win**	**soa**
TOEFL	ukwac	malt	core	basic	1	5k	simple-ll	log	cosine	100	900	rank	85.0	93.7	100
WS	wacky	malt	core	basic	4	20k	simple-ll	log	cosine	50	500	rank	0.68	0.69	0.83
AP	wacky	malt	core	basic	1	20k	simple-ll	log	cosine	0	300	rank	**0.70**	0.69	0.79
	Dependency typed														
	corpus	**parser**	**d.gr**	**d.st**	**p.len**	**c.dim**	**score**	**transf**	**metric**	**d.sk**	**r.dim**	**r.ind**	**b.synt**	**b.win**	**soa**
TOEFL	wacky	malt	core	basic	1	100k	t-score	none	cosine	100	900	rank	81.2	93.7	100
WS	ukwac	malt	core	basic	1	100k	simple-ll	log	cosine	50	700	rank	0.56	0.69	0.83
AP	wacky	malt	core	basic	1	100k	simple-ll	log	cosine	0	300	rank	**0.72**	0.69	0.79

Table 7.7: Reduced runs - best settings

Setting	**corpus**	**parser**	**d.gr**	**d.st**	**p.len**	**c.dim**	**score**	**transf**	**metric**	**d.sk**	**r.dim**	**r.ind**
Filtered, unreduced	ukwac	malt	core	basic	1	5k	simple-ll	root	cosine	–	–	rank
Filtered, reduced	wacky	malt	core	basic	2	10k	simple-ll	log	cosine	50	500	rank
Typed, unreduced	wacky	malt	core	basic	1	100k	z-score	none	cosine	–	–	rank
Typed, reduced	wacky	malt	core	basic	1	100k	simple-ll	log	cosine	50	700	rank

Table 7.8: General best settings (filtered and typed, reduced and unreduced)

Setting	**TOEFL**	**WS**	**RG**	**AP**	**BATTIG**	**MITCHELL**	**ESSLLI**
filtered, unreduced, reference	72.5	**0.69**	0.76	0.67	0.89	0.73	0.75
filtered, unreduced, general	76.2	0.54	0.75	0.67	0.89	0.73	0.75
filtered, reduced, reference	**85.0**	0.68	**0.84**	0.70	0.89	0.73	0.79
filtered, reduced, general	82.5	0.63	0.83	0.64	0.88	0.70	0.79
typed, unreduced, reference	80.0	0.53	0.74	0.68	0.77	0.75	0.73
typed, unreduced, general	80.0	0.54	0.79	0.60	0.73	0.72	0.70
typed, reduced, reference	81.2	0.56	0.74	**0.72**	0.83	0.73	**0.80**
typed, reduced, general	82.5	0.62	0.82	0.70	**0.91**	**0.80**	0.77

Table 7.9: Evaluation overview: syntax-based DSMs, word similarity datasets

7.6 Summing up

In this chapter, we have presented the results of a large-scale evaluation study of syntax-based DSMs. We showed that, even after extensive parameter tuning, syntax-based DSMs outperform comparable window-based models only in the noun clustering task – and in this case, the finer-grained dependency-typed models outperform the dependency filtered ones. The answer to the question raised in the title of this chapter is that, in general, the performance gain achieved with syntax-based DSMs may not justify the computational effort. However, in more complex tasks such as concept clustering (which can be seen as a form of mediated similarity, via the shared hypernym), syntax-based contexts do boost DSM performance. Interestingly, in this case, performance gain is higher when the syntactic information encoded in the parse trees is used in its "full power" – that is, by encoding the syntactic relation in the feature label.

We have identified many commonalities between dependency filtered and window-based DSMs. A significant core of the parameter space (metric, score, transformation, relatedness index) is common to both types of models, in terms of the impact on performance as well as best parameter values. In a dependency filtered setting, path length trades off between paradigmatic similarity and non-attributional relatedness in the same way window size does (WS requires longer syntactic paths and larger window sizes). The pattern of the number of latent dimensions to be discarded is also comparable to that of window-based models: synonymy is better modeled by discarding the first 100 SVD dimensions, the mixture of similarity and relatedness encoded in WS is better captured discarding the first 50 dimensions, and all dimensions are necessary for noun clustering. Interestingly, the picture becomes less sharp with dependency-typed models, for which even for AP discarding the first 50 dimensions appears to be a valid alternative. These results set the stage for a semantic interpretation of the ordering of the SVD-reduced dimensions – we will come back to this point in the next chapter, with more evidence coming from the direct comparison of syntagmatic and paradigmatic relations.

It is left for future work to establish to what extent our conclusions generalize to different languages. For example, DSM evaluation on German reveals a mixed picture: on the one hand, Bott & Schulte im Walde (2015) found no advantage for syntax-based models over window-based ones in a quite linguistically oriented task, namely the prediction of particle verb compositionality; on the other, Utt & Padó (2014) did find advantages in the use of syntactic information in the German counterparts of TOEFL and WS, as well as in more linguistically challenging tasks such as the prediction of thematic fit ratings.

8

Modeling syntagmatic and paradigmatic relations

The studies presented in this chapter contribute to the debate concerning the nature of the semantic representations built by DSMs, and they do so by zooming in into a well established dychotomy in DSM research, namely that between paradigmatic and syntagmatic relations. Paradigmatic relations hold between words that occur in similar contexts; they are also called relations *in absentia* (Sahlgren, 2006) because paradigmatically related words do not co-occur. Examples of paradigmatic relations are *synonyms* (e.g., *frigid–cold*) and *antonyms* (e.g., *cold–hot*). Syntagmatic relations hold between words that co-occur (relations *in praesentia*) and therefore occur in shared contexts. Typical examples of syntagmatic relations are phrasal associates (e.g., *help–wanted*) and syntactic collocations (e.g., *dog–bark*).

Distributional modeling has already dealt with the issue of the difference between paradigmatic and syntagmatic relations (e.g., Sahlgren, 2006; Rapp, 2002). In this connection, the key contributions of the study presented in this chapter are the scope of its evaluation (in terms of semantic relations and model parameters), the focus on cognitive datasets (priming and free-association norms) and the new perspective on paradigmatic vs. syntagmatic models provided by our results.

As far as the scope of the evaluation is concerned, this is the first study in which the comparison involves such a wide range of semantic relations (*paradigmatic:* synonyms, antonyms and co-hyponyms; *syntagmatic:* syntactic collocations, backward and forward phrasal associates). Moreover, our evaluation covers the large parameter space employed in chapters 6 and 7, and we consider the variation in performance achieved by different parameter settings as a cue towards characteristic aspects of specific relations (or groups of relations).

This work also differs from previous studies in its focus on second-order models (DSMs). We show that DSMs are able to capture both paradigmatic and syntagmatic relations with appropriate parameter settings. In addition, this focus provides a uniform experimental design for the evaluation. For example, parameters like window size and directionality apply to window-based DSMs and collocation lists, but not to term-context models; dimensionality reduction, whose effect has not yet been explored systematically in the context of a comparison between syntagmatic and paradigmatic relations, is not applicable to collocation lists.

The study presented in this chapter proceeds in two steps. First, we carry out an intrinsic evaluation of DSMs in a multiple-choice task based on the priming datasets introduced in section 3.4. DSM performance is evaluated here by means of the same

methodology and in the same parameter space as in the previous chapters – keeping the focus of the discussion on the parameters responsible for the distinction between syntagmatic and paradigmatic relations. Section 3.4 is based on Lapesa, Evert, & Schulte im Walde (2014). Next, we rely on the robust settings identified in the intrinsic evaluation to compare DSMs and collocation models in an extrinsic task: the prediction of free-association norms, introduced in section 3.3.1; this section is based on Lapesa & Evert (2014b). The chapter is structured accordingly: after a brief overview of previous work on the distinction between paradigmatic and syntagmatic relations in section 8.1, we proceed to the results of the intrinsic evaluation on priming datasets in section 8.2 and then move on to the extrinsic evaluation on free associations in section 8.3. We conclude by summarizing the findings and defining further research directions with respect to this topic.

8.1 Previous work

In this section we discuss previous work relevant for the distributional modeling of paradigmatic vs. syntagmatic relations. We focus only on two studies (Rapp, 2002; Sahlgren, 2006), in which the two classes of relations are compared at a global level, and not on studies that are concerned with specific semantic relations, such as *synonymy* (Edmonds & Hirst, 2002; Curran, 2003) or *hypernymy* (Weeds et al., 2004; Lenci & Benotto, 2012), with the discrimination of paradigmatic relations (Santus et al., 2016), or with the modeling of syntagmatic predicate preferences (McCarthy & Carroll, 2003; Erk et al., 2010).

In the previous studies, the comparison of syntagmatic and paradigmatic relations has been implemented in terms of an opposition between different classes of corpus-based models: term-context models (words as targets, documents or context regions as features) vs. window-based models (words as targets and features) in Sahlgren (2006); collocation lists vs. window-based models in Rapp (2002). Given the high terminological variation in the literature, in this thesis we adopt the labels *syntagmatic* and *paradigmatic* to characterize different types of semantic relations, and we will use the labels *first-order* and *second-order* to characterize corpus-based models with respect to the kind of co-occurrence information they encode. We will refer to collocation lists as *first-order models*, and to window-based DSMs as *second-order models*.

Rapp (2002) integrates first-order (co-occurrence lists) and second-order (window-based DSMs) information to distinguish syntagmatic and paradigmatic relations. Under the assumption that paradigmatically related words will be found among the closest neighbors of a target word in the DSM space and that paradigmatically and syntagmatically related words will be intermingled in the list of collocates of the target word, Rapp proposes to exploit the comparison of the most salient collocates and the nearest DSM neighbors to distinguish between the two types of relations. Sahlgren (2006) compares term-context and bag-of-words DSMs in a number of tasks involving syntagmatic and paradigmatic relations. His evaluation covers several tasks. The first step is a comparison between the thesaurus entries for target words (containing both paradigmatically and syntagmatically related words) and neighbors in the distributional spaces: while term-context DSMs produce both syntagmatically and paradigmatically related words, the nearest neighbors in a bag-of-words DSM mainly provide paradigmatic information. Window-based DSMs also perform better than term-context models in predicting

Relation	Dataset	Unreduced	Reduced
Syntagmatic	GEK	93.0	86.6
Syntagmatic	FPA	89.6	79.2
Syntagmatic	BPA	88.0	76.9
Paradigmatic	SYN	92.4	84.7
Paradigmatic	COH	88.5	75.1
Paradigmatic	ANT	88.7	75.8

Table 8.1: Multiple choice on priming datasets: adjusted R^2

association norms, in the TOEFL multiple-choice synonymy task and in the prediction of antonyms (although the difference in performance is less significant here). Last, word neighborhoods are analysed in terms of their part-of-speech distribution: Sahlgren (2006) observes that window-based spaces contain more neighbors with the same part of speech as the target than term-context spaces: his conclusion is that window-based spaces priviledge paradigmatic relations, based on the assumption that paradigmatically related word pairs belong to the same part of speech, while this is not necessarily the case for syntagmatically related word pairs.

Summing up, in both Rapp (2002) and Sahlgren (2006) it is claimed that second-order models perform poorly in predicting syntagmatic relations. However, neither of these studies involves datasets containing *exclusively syntagmatic relations*, as the evaluation focuses either on paradigmatic relations (TOEFL multiple choice test, antonymy test) or on resources containing both types of relations (thesauri, association norms). The studies presented in this chapter aim at filling this gap.

8.2 Multiple choice task on priming datasets

In this section, we discuss the results of the evaluation of DSMs in the multiple-choice task on the priming datasets described in section 3.4. This evaluation task tests the accuracy of DSMs in telling which of the two candidate words is the *consistent prime* based on the distributional representation of the two candidate primes (*bark*, *meow*) and the target (*dog*). Note that in this task our use of neighbor rank mirrors the experimental setting, as we calculate the position of the target among the neighbors of the two candidate primes.

We selected three syntagmatic datasets: the Generalised Event Knowledge (GEK) dataset as well as the forward and backward phrasal associates from SPP (FPA, BPA). As for paradigmatic relations, we selected the synonyms (SYN), antonyms (ANT) and cohyponyms (COH) from SPP. For more details and considerations on the task and the datasets refer to section 3.4. The DSMs evaluated in chapter 6 were tested on each priming dataset separately. Table 8.1 lists the model fit on each dataset, for the reduced and unreduced runs. Despite some variability across relations and between unreduced and reduced runs, the R^2 values are always high showing that the linear model explains a large part of the observed performance. We also note the familiar drop in model fit from unreduced to reduced runs, and a lower fit for smaller datasets (FPA, BPA, COH, ANT).

Let us now turn to the discussion of the feature ablation displayed in figures 8.1 and 8.2 for the paradigmatic relations and 8.3 and 8.4 for the syntagmatic ones. Like in the previous chapters, the feature ablation plots are complemented by the corresponding

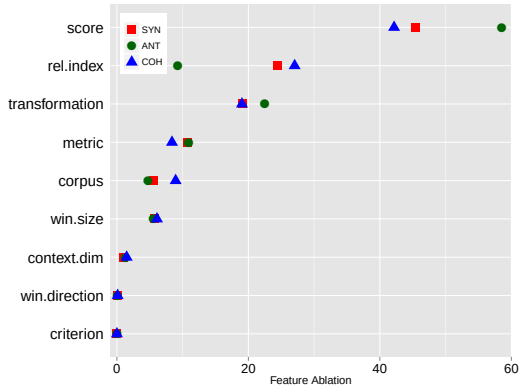

Figure 8.1: Paradigmatic, unreduced

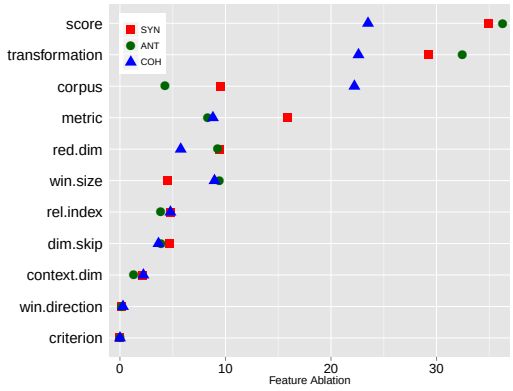

Figure 8.2: Paradigmatic, reduced

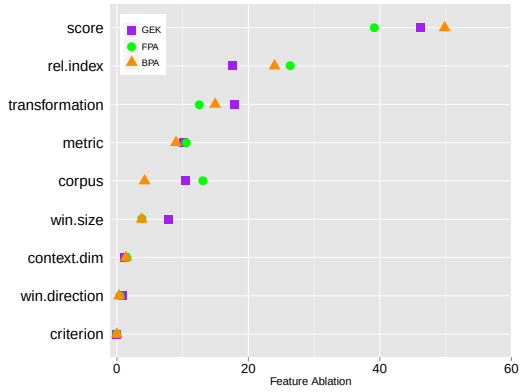

Figure 8.3: Syntagmatic, unreduced

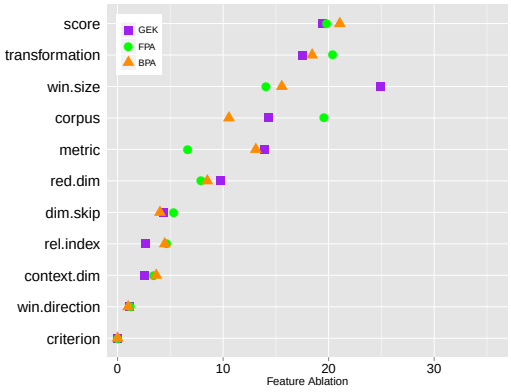

Figure 8.4: Syntagmatic, reduced

interaction tables (tables 8.2 to 8.5).

Our observations concerning the explanatory power of different parameters can be summarized in the following points:

- *Feature score* and *feature transformation* are consistently crucial in determining DSM performance, both in reduced and unreduced runs, and for both paradigmatic and syntagmatic relations. At this point in this dissertation, this is not a surprising result and it confirms the trend we are already familiar with.

- The *index of distributional relatedness* plays a substantial role in determining model performance, more so than in the standard word similarity tasks (in particular with respect to the other multiple choice task, TOEFL). The gain in explanatory power is reflected in the interaction tables, where we find more cases in which the effect of the relatedness index modulates in combination with other parameters (its interaction with feature score being particularly strong).

- SVD parameters play a significant role, but to a lesser extent as compared to the word similarity tasks.

	SYN	ANT	COH
score × transf	7.74	10.85	8.62
score × rel.index	5.62	2.75	7.25
transf × metric	1.13	2.35	1.30
window × score	0.81	1.90	1.33
score × metric	0.71	1.64	0.75
corpus × score	0.52	0.99	0.97
window × transf	–	0.79	0.70
window × rel.index	–	–	0.87
score × cont.dim	–	–	0.82
transf × rel.index	–	–	0.65
metric × cont.dim	–	0.56	–

Table 8.2: Paradigmatic, unreduced

	SYN	ANT	COH
score × transf	15.65	16.73	9.92
window × transf	1.95	3.58	3.33
metric × red.dim	1.54	1.48	1.26
window × score	0.89	1.26	1.44
score × metric	1.45	1.19	0.70
score × dim.skip	1.09	0.71	1.08
window × dim.skip	0.58	1.00	0.97
transf × dim.skip	0.92	0.61	0.80
score × cont.dim	0.78	0.73	0.76
corpus × window	–	0.68	1.38
metric × rel.index	0.65	0.93	–
corpus × score	0.66	–	0.79
score × rel.index	0.54	–	0.72
metric × cont.dim	0.97	–	–
corpus × metric	–	–	0.69
transf × metric	–	–	0.58
transf × rel.index	–	0.52	–

Table 8.3: Paradigmatic, reduced

	GEK	FPA	BPA
score × rel.index	5.43	6.05	7.14
score × transf	6.69	4.35	6.13
transf × metric	0.77	1.34	1.08
window × transf	1.46	1.33	–
score × metric	0.61	0.81	–
corpus × window	0.57	–	0.85
corpus × score	0.71	–	–
corpus × rel.index	–	–	0.70
window × score	–	0.69	–
metric × cont.dim	–	0.56	–
score × metric	–	–	0.54

Table 8.4: Syntagmatic, unreduced

	GEK	FPA	BPA
score × transf	7.47	8.15	8.10
window × transf	2.32	3.20	1.46
corpus × window	1.15	0.75	2.30
score × dim.skip	1.20	1.30	1.04
metric × red.dim	2.02	0.77	0.94
score × metric	0.97	1.01	0.99
metric × cont.dim	1.36	0.72	0.76
corpus × metric	0.94	0.55	0.59
metric × dim.skip	0.88	0.89	–
transf × dim.skip	0.77	–	0.76
metric × rel.index	–	0.58	0.76
window × score	0.51	0.79	–
corpus × transf	–	–	0.71
score × rel.index	–	–	0.53
corpus × score	0.51	–	–
score × cont.dim	0.51	–	–

Table 8.5: Syntagmatic, reduced

- For both syntagmatic and paradigmatic relations, *source corpus* gains explanatory power in a SVD-reduced setting. The same holds for the *size of the context window*, but only with respect to paradigmatic relations.
- Within paradigmatic relations, we note a significant drop in explanatory power for the *relatedness index* when it comes to antonyms. Within syntagmatic relations, the *size of the context window* appears to be more crucial for the GEK dataset than it is for FPA and BPA. In the next section, the analysis of the best choices for this parameter will provide a clue for the interpretation of these differences.
- Three parameters have little to no explanatory power: *directionality of the context window*, *criterion for context selection* and *number of context dimensions*.

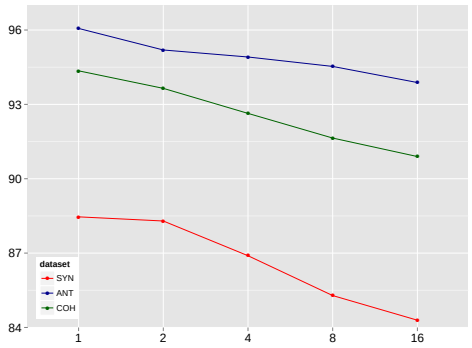

Figure 8.5: Window, paradigmatic, unred

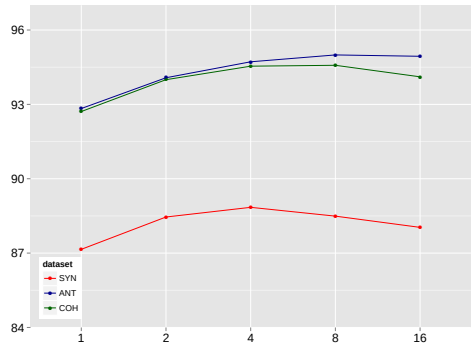

Figure 8.6: Window, paradigmatic, red

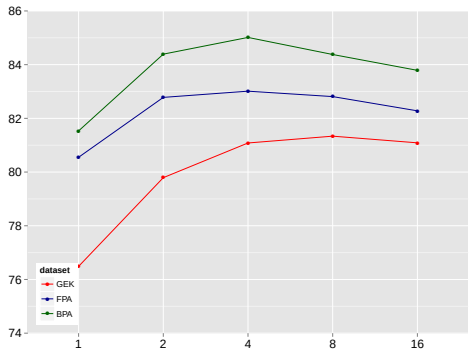

Figure 8.7: Window, syntagmatic, unred

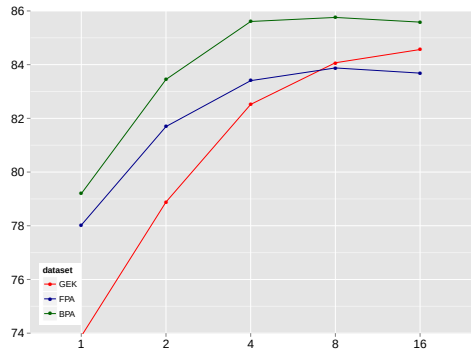

Figure 8.8: Window, syntagmatic, red

8.2.1 Best parameter values

In this section, we identify the best parameter values for our datasets, keeping the focus of the discussion on the parameters which contribute to the comparison between syntagmatic and paradigmatic relations. In this connection, relevant parameters are *window size*, *relatedness index*, and *dimensionality reduction* parameters. We will address them first and discuss in detail our interpretation of the semantic import of the different parameter values, comparing reduced and unreduced runs directly (when applicable).

Window size This parameter plays a crucial role in contrasting syntagmatic and paradigmatic relations, as well as different relations within those general groups. The plots in figures 8.5 and 8.6 display its partial effect for paradigmatic relations in the unreduced and reduced settings, respectively. The plots in figures 8.7 and 8.8 display its partial effect for syntagmatic relations.

According to figure 8.5, when no dimensionality reduction is involved, a very small context window (i.e., one word) is sufficient for all paradigmatic relations, and DSM performance decreases as soon as we enlarge the context window. Interactions between *window size* and *score* and *transformation* (cf. supplementary material) help us getting a better characterization of this effect (because optimal score/transformation settings, discussed below, allow for slightly larger windows): robust window sizes are 2 for SYN and COH, and 4 for ANT. The picture changes when applying dimensionality reduction:

a 4-word window is a robust choice for all paradigmatic relations (although ANT shows a further increase in performance with an 8-word window), even in the SYN task that is traditionally associated with very small windows of 1 or 2 words (Sahlgren, 2006).

A significant interaction between window size and number of skipped dimensions (not shown here for reasons of space) sheds further light on this matter. Without skipping SVD dimensions, the reduced models achieve optimal performance for a 2-word window and degrade more (COH) or less (ANT) quickly for larger windows. With 50 or 100 dimensions skipped, performance improves up to a 4- or 8-word window. Our interpretation of this fact is that the first SVD dimensions capture general domain and topic information dominating the co-occurrence data; removing these dimensions reveals paradigmatic semantic relations even for larger windows.

Figure 8.7 shows that for syntagmatic relations without dimensionality reduction a larger context window of 4 words is needed for FPA and BPA; a further increase of the window is detrimental. For the GEK dataset, performance peaks at 8 words, and decreases only minimally for even larger windows. An inspection of the interaction plots involving *window size* confirms these robust choices. As before, dimensionality reduction improves performance for large co-occurrence windows. For FPA and BPA, the optimum is achieved with a window of 4–8 words; performance on GEK continues to increase up to a window of 16 words, the largest window size considered in our experiments.

Overall, the observed patterns reflect differences in the nature of the semantic relations involved: smaller windows provide better contextual representations for paradigmatic relations, while larger windows are needed to capture syntagmatic relations with bag-of-words DSMs, because co-occurring words share a large portion of their context windows. Intermediate window sizes are sufficient for phrasal collocates (which are usually adjacent), while event-based relatedness (GEK) requires larger windows.

Returning briefly to the slight preference shown by ANT for a larger window, we notice that ANT seems to be more similar to the syntagmatic relations than SYN and COH. This is in line with the observations of Justeson & Katz (1992) concerning the tendency of antonyms to co-occur (e.g., in coordinations such as *short and long*). Antonyms appear to be the least canonical among the paradigmatic relations: like synonyms, antonyms are interchangeable, but (a) they enter into syntagmatic patterns that are uncommon for synonyms and (b) they may also introduce a topic shift (e.g., *happy/sad*).

Dimensionality reduction We now focus on the parameters related to dimensionality reduction, namely the *number of latent dimensions* (figures 8.9 and 8.10) and the *number of skipped dimensions* (figures 8.11 and 8.12).

We have found no difference between syntagmatic and paradigmatic relations with respect to the *number of latent dimensions*: the more, the better in both cases (900 dimensions). The *number of skipped dimensions*, however, shows some variability across the different relations. The results for SYN are in agreement with our own findings (and those of Bullinaria & Levy (2012)) on TOEFL: skipping 50 or 100 initial dimensions improves performance. Skipping dimensions makes minimal difference for COH (best choice is 50 dimensions), while the full range of reduced dimensions is necessary for ANT. Within syntagmatic relations, the full range of latent dimensions ensures good performance on phrasal associates (even if skipping 50 dimensions is not detrimental for

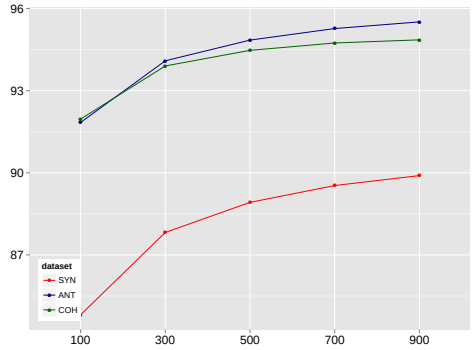

Figure 8.9: Reduced dim., paradigm.

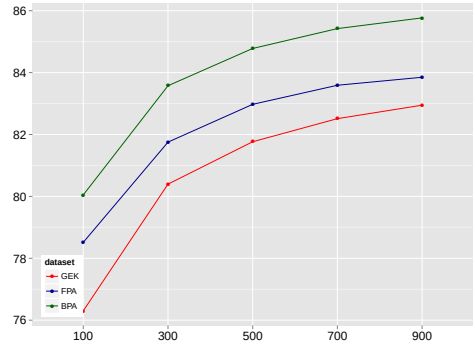

Figure 8.10: Reduced dim., syntagm.

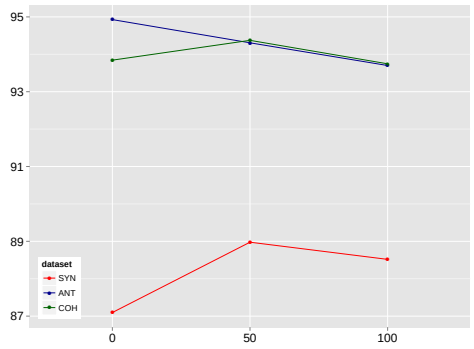

Figure 8.11: Skipped dim., paradigm.

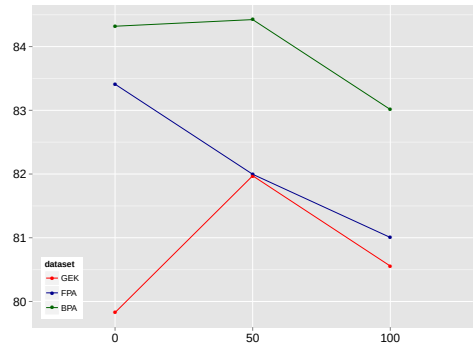

Figure 8.12: Skipped dim., syntagm.

BPA). GEK shows a pattern similar to SYN, with 50 skipped dimensions leading to a considerable improvement.

Relatedness Index As shown in figure 8.13 for the unreduced runs and in figure 8.14 for the reduced runs, *neighbor rank* is consistently better than *distance* on all datasets. The plots display the main effect because this allows a straightforward comparison between datasets. An inspection of interactions involving *relatedness index* confirms that rank is the best parameter in all cases, and the high ablation value comes from the fact that, as we have observed in the previous chapters, its effect modulates differently in relation to other parameter values (e.g., bigger difference with Manhattan, smaller difference with association measures). This is not surprising because our use of neighbor rank captures asymmetry and mirrors the experimental setting, in which targets are shown after primes. A further observation may be made in relation to the degree of asymmetry of different relations. In particular, the unreduced setting shows that syntagmatic relations are subject to stronger asymmetry effects than the paradigmatic ones, presumably due to the directional nature of the relations involved (phrasal associates and syntactic collocations). Among paradigmatic relations, antonyms appear to be the least asymmetric ones, because using neighbor rank instead of distance makes a comparatively small difference.

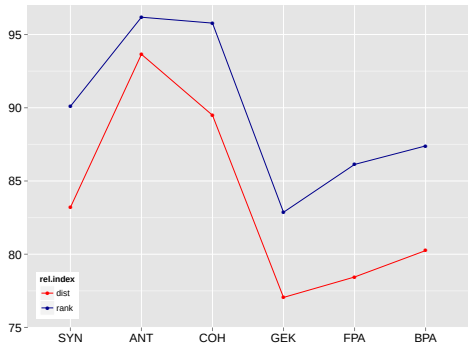

Figure 8.13: Relatedness index, unreduced

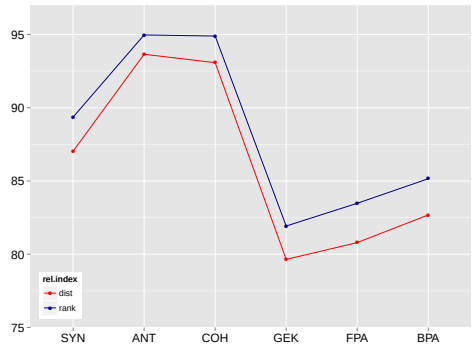

Figure 8.14: Relatedness index, reduced

Remaining parameters A very strong interaction between *score* and *transformation* characterizes all settings; the corresponding plots are shown in figures 8.15 to 8.26.

As already observed in the previous chapters, in the unreduced runs, untransformed vectors in combination with association measures based on significance tests (simple-ll, t-score, z-score) are better than Dice and, to a lesser extent, MI. Overall, we observe the familiar shift from unreduced to reduced runs: no transformation is the robust choice in the unreduced runs, while root is the robust choice in the reduced runs; simple-ll requires more aggressive de-skewing, as log overscores root transformation in the reduced runs. More specifically, in the unreduced runs, *untransformed z-score* is the best choice for all datasets (on the paradigmatic ones, though, root-transformed simple-ll is very competitive as well). In the reduced runs, *simple-ll* is the best choice in combination with a *logarithmic transformation* for the paradigmatic relations, whereas *z-score* appears to be the best measure for syntagmatic relations in combination with a *root transformation*, with minimal differences from untransformed MI and log-transformed simple-ll. *Log-transformed simple-ll* is thus the most robust parameter for the SVD runs.

The optimal *metric* is *cosine* distance, consistently outperforming *manhattan* in both unreduced (figure 8.27) and reduced (figure 8.28) runs.

As far as *source corpus* is concerned (figure 8.29 and 8.30), BNC consistently yields the worst results, while *WaCkypedia* and *ukWaC* appear to be almost equivalent in the unreduced runs. The trade-off between quality and quantity appears to be strongly biased towards sheer corpus size in the case of distributional models. For syntagmatic relations and SVD-reduced models, *ukWaC* is clearly the best choice. This suggests that syntagmatic relations are better captured by features from a larger lexical inventory, combined with the abstraction performed by SVD.

The inspection of partial effect plots for minimally explanatory parameters supports the choice of unmarked default values for *directionality of the context window* (*undirected*) and *criterion for context selection* (*frequency*), as well as an intermediate *number of context dimensions* (*50000* dimensions).

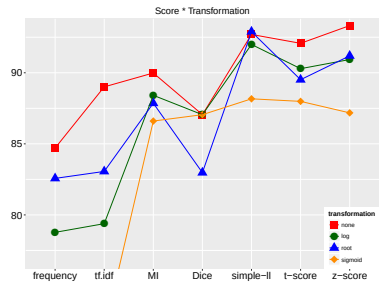

Figure 8.15: SYN, unred

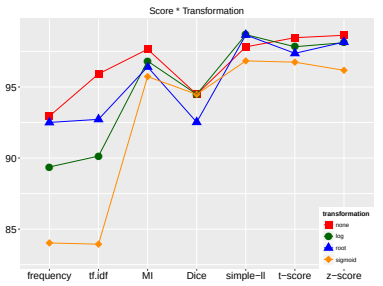

Figure 8.16: ANT, unred

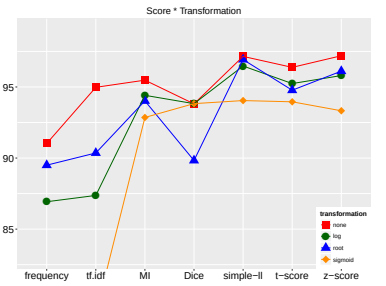

Figure 8.17: COH, unred

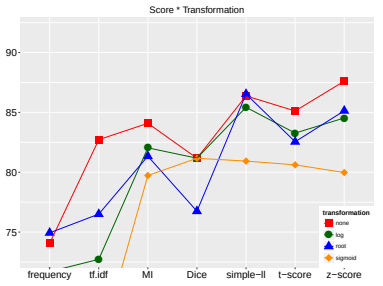

Figure 8.18: GEK, unred

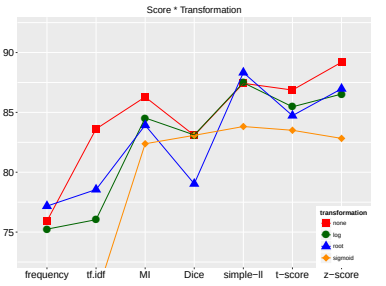

Figure 8.19: FPA, unred

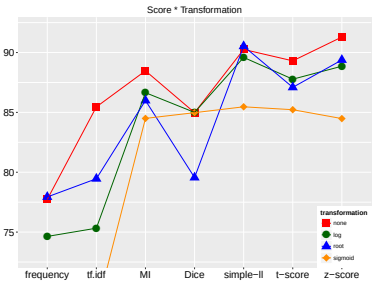

Figure 8.20: BPA, unred

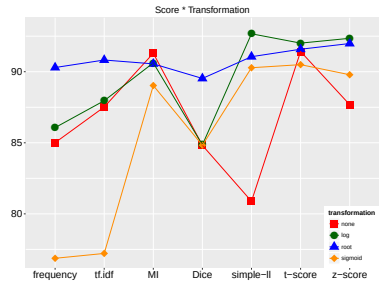

Figure 8.21: SYN, red

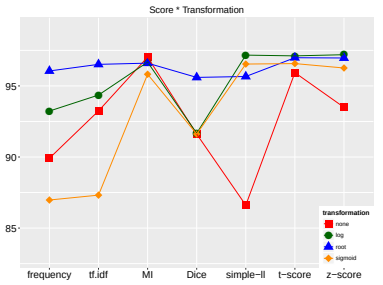

Figure 8.22: ANT, red

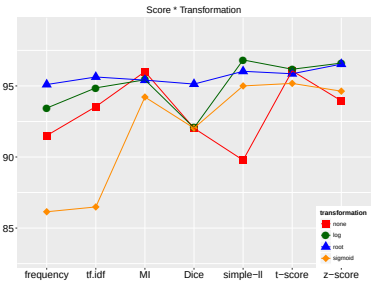

Figure 8.23: COH, red

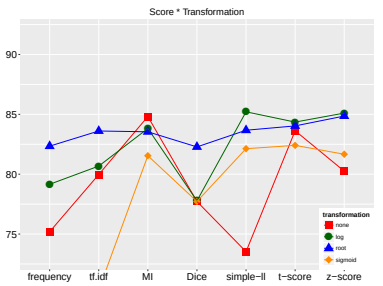

Figure 8.24: GEK, red

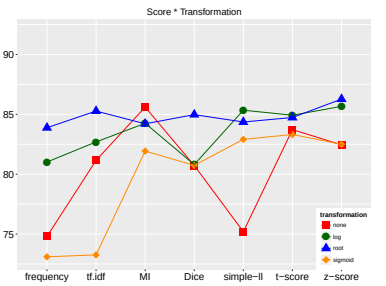

Figure 8.25: FPA, red

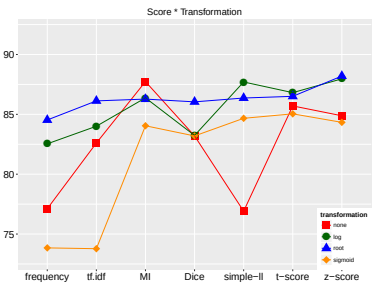

Figure 8.26: BPA, red

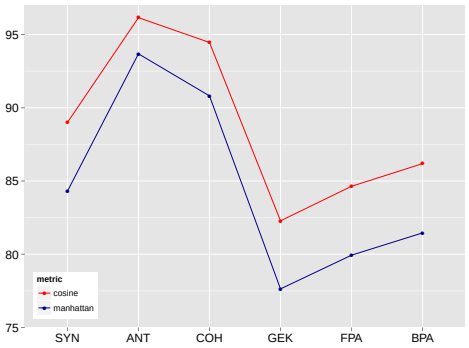

Figure 8.27: Distance metric, unreduced

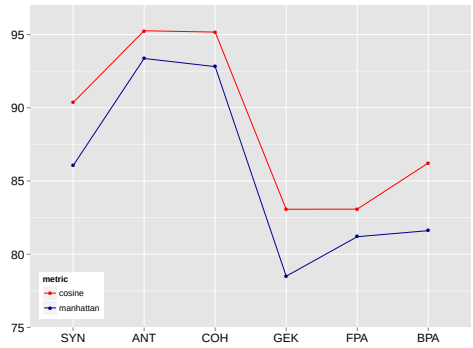

Figure 8.28: Distance metric, reduced

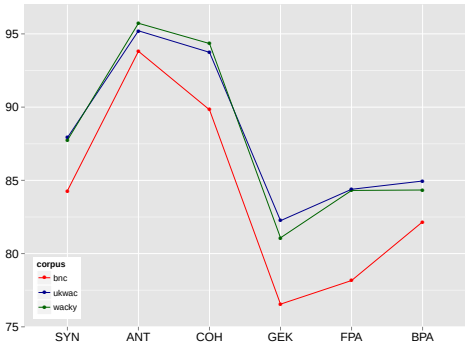

Figure 8.29: Corpus, unreduced

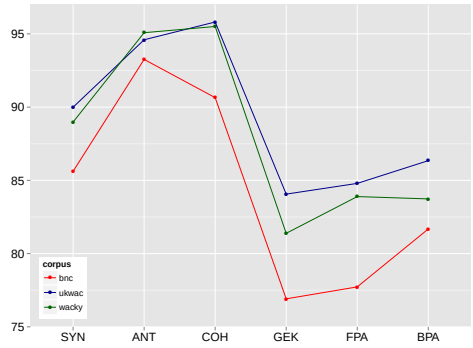

Figure 8.30: Corpus, reduced

8.2.2 Best settings

We conclude by comparing the performance achieved by our robust choice of optimal parameter values identified in section 8.2.1. Tables 8.6 and 8.7 display the best parameter settings for each dataset along with their accuracy.

As a next step, we identified parameter combinations that work well for all types of syntagmatic (*Best Syntagmatic*) and paradigmatic relations (*Best Paradigmatic*), as well as an even more general setting (*Best Priming*) that is suitable for paradigmatic and syntagmatic relations alike. Such best settings are shown in table 8.8, while their performance on each dataset is reported in tables 8.9 for the unreduced runs and 8.10 for the reduced runs. For comparison, we also report the performance of the best general settings identified in chapter 6 on the word similarity tasks.

A high-level inspection of the performance of the different settings on the different semantic relations in our datasets allows us to make a number of observations:

- Our methodology allowed us to identify robust settings, which come close to the performance of the best run (cf. best models in appendix C).
- Syntagmatic relations constitute a more difficult task. This is not surprising, given that for some of the pairs in the dataset primes and targets belong to different

parts of speech and (a) in SPP, it can happen that the inconsistent prime matches the consistent one in part of speech, while the consistent one does not, (b) in GEK primes belong to the same part of speech, but inconsistent primes have been carefully sampled, which makes them challenging distractors.

- SVD does not improve performance, at least not to the same extent as we have observed in the word similarity tasks.
- ANT confirms its special status among the paradigmatic relations. The DSMs that achieve better performances on it are the syntagmatic ones or, among the word similarity best settings, those that have been tuned on Ratings and Noun Clustering (and not on TOEFL).
- On the paradigmatic relations, the settings that have been tuned on the word similarity tasks outperform (or come very close to) the dedicated settings. On the syntagmatic relations, on the other hand, the parameter values tuned in the dedicated experiments outperform the word similarity settings. This is not surprising, given that such datasets do not target syntagmatically related pairs (with the exception of the relatedness subset of WS).

setting	corpus	win	dir	orig.dim	crit	score	transf	metric	rel.ind	acc
SYN	wacky	2	undir	20000	f	z-score	none	cosine	rank	94.72
ANT	wacky	4	undir	20000	f	simple-ll	root	cosine	rank	100.00
COH	wacky	2	undir	20000	f	z-score	none	cosine	rank	99.34
FPA	wacky	4	undir	50000	f	z-score	none	cosine	rank	94.44
BPA	ukwac	4	undir	50000	f	z-score	none	cosine	rank	97.75
GEK	ukwac	8	undir	50000	f	z-score	none	cosine	rank	94.31

Table 8.6: Best settings, unreduced runs: datasets, parameter values, accuracy

setting	corpus	win	dir	orig.dim	crit	score	transf	n.dim	d.skip	metric	rel.ind	acc
SYN	wacky	4	undir	50000	f	simple-ll	log	900	50	cosine	rank	96.56
ANT	wacky	8	undir	50000	f	simple-ll	log	900	0	cosine	rank	100.00
COH	ukwac	4	undir	50000	f	simple-ll	log	900	50	cosine	rank	98.68
FPA	ukwac	8	undir	50000	f	z-score	root	900	0	cosine	rank	93.06
BPA	ukwac	8	undir	50000	f	z-score	root	900	0	cosine	rank	95.51
GEK	ukwac	16	undir	50000	f	z-score	root	900	50	cosine	rank	95.30

Table 8.7: Best settings, reduced runs: datasets, parameter values, accuracy

setting	corpus	win	dir	orig.dim	crit	score	transf	n.dim	dim.skip	metric	rel.ind
Paradigmatic, unr	wacky	2	undir	20000	f	z-score	none	–	–	cosine	rank
Syntagmatic, unr	wacky	4	undir	50000	f	z-score	none	–	–	cosine	rank
General, unr	ukwac	2	undir	50000	f	z-score	none	–	–	cosine	rank
Paradigmatic, red	ukwac	4	undir	50000	f	simple-ll	log	900	50	cosine	rank
Syntagmatic, red	ukwac	8	undir	50000	f	z-score	root	500	0	cosine	rank
General, red	ukwac	4	undir	50000	f	simple-ll	log	900	0	cosine	rank

Table 8.8: General best settings

	SYN	ANT	COH	FPA	BPA	GEK
Best Paradigmatic	94.72	99.26	99.34	90.28	91.01	90.35
Best Syntagmatic	96.56	**100.00**	98.68	**97.92**	**97.75**	**94.06**
Best Priming	96.79	99.26	98.68	95.14	95.51	92.57
Best TOEFL	95.41	98.52	98.01	94.44	95.51	91.09
Best Ratings	94.50	**100.00**	99.34	95.14	93.26	90.84
Best Clustering	94.95	**100.00**	99.34	95.14	93.26	90.84
Best Word Similarity	94.95	**100.00**	99.34	95.14	93.26	90.84
Best PPMI	93.81	99.26	98.68	90.97	91.01	89.60
Best PPMI+	**97.25**	**100.00**	**100.00**	93.06	91.01	89.85
Best Cognitive	91.51	95.56	96.69	83.33	88.76	80.94

Table 8.9: General best settings, unreduced - comparison to best settings from chapter 6

	SYN	ANT	COH	FPA	BPA	GEK
Best Paradigmatic	96.33	99.26	98.68	90.28	**97.75**	**94.80**
Best Syntagmatic	94.50	99.26	99.34	**93.06**	94.38	92.08
Best Priming	96.33	99.26	98.68	91.67	95.51	91.34
Best TOEFL	94.95	95.56	97.35	81.94	87.64	86.88
Best Ratings	96.56	99.26	**100.00**	84.72	95.51	93.07
Best Clustering	95.18	**100.00**	98.68	91.67	92.13	90.84
Best Word Similarity	96.56	99.26	99.34	87.50	93.26	90.84
Best PPMI	95.18	99.26	99.34	90.28	92.13	91.58
Best PPMI+	**97.25**	**100.00**	99.34	87.50	96.63	93.32
Best Cognitive	89.91	91.85	96.69	84.03	88.76	81.19

Table 8.10: General best settings, reduced - comparison to best settings from chapter 6

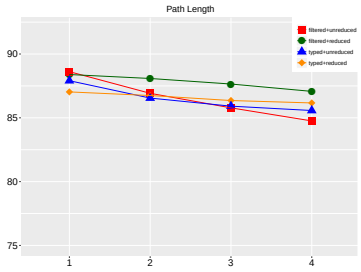

Figure 8.31: SYN

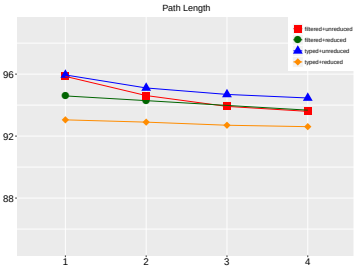

Figure 8.32: ANT

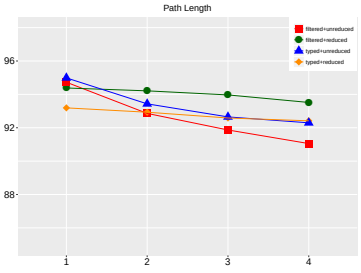

Figure 8.33: COH

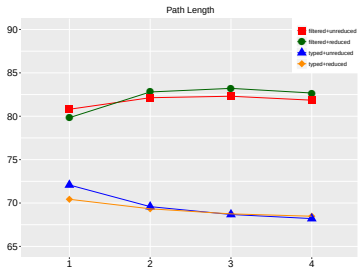

Figure 8.34: FPA

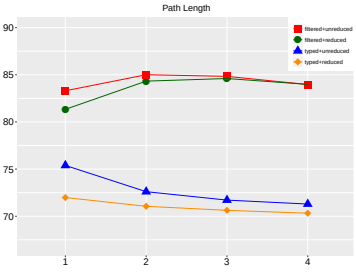

Figure 8.35: BPA

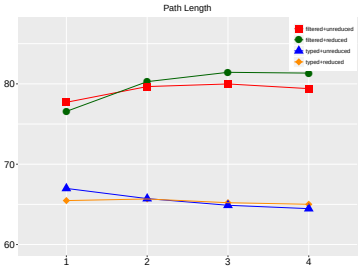

Figure 8.36: GEK

8.2.3 Dependency-based models

In this section, we provide a quick overview of the results of the dependency-based experiments. While we do not elaborate on details, as we have done in chapter 7 for word similarity tasks, there are still a number of interesting observations to be made, in particular in comparison with the window-based results presented in this chapter.

The feature ablation experiments (see the relevant plots in the supplementary material) follow the general tendencies identified in this dissertation. *Score* and *transformation* play a strong role in determining model performance, and so does *distance metric*. Best parameter values show no unexpected results: association measures are the best, and so is cosine.

Relatedness index is more powerful here than in standard word similarity experiments. This is expected, given that it is exactly in this task that the use of neighbor rank mirrors the psycholinguistic setup. Relatedness index is more powerful in the sparser typed and unreduced setting than it is in the dependency filtered one, and it loses power when SVD comes into play; however, it always stays in the middle range of the parameter ranking. Neighbor rank outperforms distance across the board.

Source corpus has also a strong impact on the performance, for syntagmatic and paradigmatic relations alike. Larger corpora have to be preferred also in this case.

Path length, from the middle-range of the parameter ranking in the unreduced runs, gains power with SVD where syntagmatic relations are concerned (farthest contexts being necessary), while it further loses power with paradigmatic relations (closest contexts in the dependency graph suffice in this case). The effect plots for *path length* in the four settings are displayed in figures 8.31 to 8.36. In the dependency filtered setting, syntagmatic relations exhibit a preference for longer paths, which mirrors the preference for larger windows discussed in the window-based experiments. Among paradigmatic

relations, there is a consistent preference for *shorter paths* – even for the antonyms, whose "special" status as a paradigmatic relation has already been discussed before. In the dependency typed experiments a path of length 1 is the best parameter value for all relations (and not even SVD reduction manages to make sense of co-occurrence matrices that are built with longer dependency paths).

SVD parameters have limited impact, in particular the *number of skipped dimensions*. Best values follow familiar trends (SYN: 50, 0 for ANT and COH, 0 or 50 for the syntagmatic relations), though we notice a general drop in the success of the strategy of skipping the first dimensions – probably indicating that SVD does not manage to produce a sharp picture and to spread information across the different dimensions "meaningfully". As far as *number of reduced dimensions* is concerned, the more the better, as usual.

Parsing parameters have a very limited impact on DSM performance. Specifically for the syntagmatic datasets, however, we do observe a few interactions above 0.5 R^2 which involve *parser*, *dependency style* and *dependency group*. While the inspection of the corresponding interaction plots confirms our default choices (Malt parser outperforms Stanford; basic dependencies outperform the CCprocessed ones), it is interesting that the manipulation of these parameters becomes more influential when we test the capability of the DSMs to estimate a sort of "collocational fit" for FPA and BPA, and what can be straightforwardly interpreted as thematic fit for GEK. Syntagmatic relations also consistently display an interaction between *path length* and *dependency group* (core vs. external dependencies), which characterizes them as compared to the other datasets in this study: in this case external dependencies have to be preferred, in combination with middle-range paths (while in other case we relied on core dependencies).

Performance of word similarity best settings Instead of fine-tuning the dependency models on our priming datasets, we have decided to employ them as a test set to evaluate the robustness of the word similarity settings established in chapter 7. Therefore, we wrap up this section with table 8.11, which displays the accuracy of the best settings identified in chapter 7 (refer to tables 7.6, 7.7, and 7.8 for the specific parameter values).

Despite the fact that none of the displayed syntax-based settings beats the corresponding best window-based models identified in section 8.2.2, there are still some interesting observations to make. While for the paradigmatic relations the typed models outperform the filtered ones, the syntagmatic relations show the opposite pattern (filtered better than typed). This clearly results from the fact that syntagmatic datasets contain pairs of words which can belong to different parts of speech and the use of non-part-of-speech disambiguated lemmas as targets alleviates this problem only minimally. It is also interesting that the best GEK dependency model is the one tuned on WS and is therefore sensitive to both relatedness and similarity. Finally, ANT and COH, semantically more complex compared to the other paradigmatic relation (SYN) benefit from the employment of dependency typed models. Once again, this is evidence for the fact that the best use of dependencies when dealing with more complex semantic issues is the most fine-grained one, encoded in the dependency labelled contexts.

Setting	SYN	ANT	COH	FPA	BPA	GEK
filtered, unreduced, TOEFL	92.8	97.03	95.4	**89.6**	**92.1**	85.6
filtered, unreduced, WS	93.8	97.03	98.0	85.4	87.6	80.1
filtered, unreduced, AP	94.5	**100**	98.0	88.9	89.9	83.4
filtered, unreduced, general	94.5	97.03	98.0	88.9	89.9	83.4
filtered, reduced, TOEFL	94.0	96.3	98.0	81.2	88.8	81.2
filtered, reduced, WS	94.2	98.5	98.7	86.8	87.6	**89.3**
filtered, reduced, AP	92.7	**100**	98.0	88.8	88.8	79.7
filtered, reduced, general	**94.9**	98.5	98.7	86.8	84.2	87.3
typed, unreduced, TOEFL	94.5	**100**	**100**	81.9	85.4	78.0
typed, unreduced, WS	94.7	**100**	**100**	81.2	83.1	76.0
typed, unreduced, AP	94.0	**100**	**100**	82.6	82.0	76.0
typed, unreduced, general	94.5	**100**	**100**	81.9	85.4	78.0
typed, reduced, TOEFL	91.0	97.7	98.7	76.3	78.6	69.0
typed, reduced, WS	90.8	94.0	97.3	81.9	88.7	76.4
typed, reduced, AP	92.4	98.5	99.3	82.6	80.9	77.7
typed, reduced, general	92.4	97.0	98.0	82.6	79.8	76.4

Table 8.11: Evaluation overview: syntax-based DSMs, multiple-choice priming datasets

8.3 Reverse free association task

In this section we turn to the extrinsic evaluation of the best DSM settings identified on the multiple-choice task for the priming datasets. We focus on window-based DSMs because of their performance in our development tasks, and also because of the focus of the study presented in this chapter: the comparison between collocation lists (first-order models) and DSMs (second-order models) in the reverse free association task. In this perspective, the introduction of syntactic information in the collection of collocations – albeit interesting and promising – is a follow-up step once a first understanding of the window-based dynamics at work in free association tasks has been reached.

This section introduces *NaDiR* (Naive Distributional Response generation), a corpus-based system designed for the *reverse association task*, which participated in the CogALex Shared task 2014 (Lapesa & Evert, 2014b). NaDiR is naive because it is based on a very simple algorithm that operationalizes the multiword association task as a ranking problem: candidate words from a large vocabulary are ranked by their average statistical association or distributional similarity to a given set of stimuli, then the highest-ranked candidate is selected as NaDiR's response. One advantage of this ranking approach is that it provides additional insights into the experimental results: if the model prediction is not correct, the rank of the correct answer can be used as a measure of how "close" the model came to the human associations.

The shared task datasets are derived from the Edinburgh Associative Thesaurus[1] (henceforth, EAT), which contains free associations to approximately 8000 English cue words. For each cue (e.g., *visual*) EAT lists all associations collected in the survey (e.g., *aid*, *eyes*, *aids*, *see*, *eye*, *seen*, *sight*, etc.) sorted according to the number of subjects who responded with the respective word. The CogALex shared task on multiword association is based on the EAT dataset, and is in fact a *reverse association task* (Rapp, 2014).

[1] `http://www.eat.rl.ac.uk/`

The top five responses for a target word are provided as stimuli (e.g., *aid*, *eyes*, *aids*, *see*, *eye*), and the participating systems are required to generate the original cue as a response (e.g., *visual*). The training and the test sets are random extracts of 2000 EAT items each, with minimal pre-processing (only items containing multiword units and non-alphabetical characters have been discarded).

A key problem we had to deal with while developing our system was the unrestricted set of possible responses in combination with a discrete association task, which requires the algorithm to pick exactly the right answer out of tens of thousands of possible responses. This feature makes this task much more difficult than the multiple-choice tasks often used to evaluate distributional semantic models. The problem is further complicated by the fact that the response may be an inflected form and only a prediction of the exact form was accepted as a correct answer. The need for a solution to these issues motivates various aspects of the NaDiR algorithm, which we describe in appendix D. For the purpose of this dissertation, we focus on the task of generating the correct lemma response.

Previous work on this task showed that co-occurrence models outperform distributional semantic models, and that using rank measures improves performance because it accounts for the directionality of the association (e.g., the association from stimulus to response may be larger than the association from response to stimulus). Our results corroborate both claims.

8.3.1 Experimental setup

To generate a response for a set of stimuli, we apply the following procedure:

1. For each set of stimuli, we compute association strengths or similarities between each stimulus and each response candidate in the vocabulary, adopting one of the measures described later in this section;

2. From the set of potential responses, we restrict the vocabulary to the words whose POS agrees with the predictions of the classifier described in appendix D. Stimulus words are discarded from the potential answers;

3. We compute the average association strength or similarity across all five stimuli; if a stimulus does not appear in the model, it is simply omitted from the average;

4. The top-ranked candidate is the lemma suggested as a response by NaDiR.

Corpus and vocabulary As a source corpus for the experiments presented in this section, we selected UkWaC, which was preferred to WaCkypedia because of the larger vocabulary coverage and because our window-based experiments never showed a detrimental effect for the larger corpus (and in some cases, even mild improvements).

As discussed before, the response generation strategy implemented in our experiments is based on lemmatized words. To build our lemmatized models, we relied on the linguistic annotation available with the original version of UkWaC (pos-tagging and lemmatization performed with Tree Tagger), hence relying on the same pre-processing pipeline of the window-based experiments presented in the rest of the thesis.

We restricted our vocabulary to (lemmatized) open-class words and, to keep the computational complexity manageable, we applied a frequency threshold. To estimate the

coverage of the vocabulary – and also as pre-processing for further steps in the pipeline – we performed a heuristic out-of-context lemmatization with a simple mapping strategy based on the linguistic annotation already available in UkWaC (see appendix D for more details). We believe that the advantages of constructing distributional models based on lemmatized words overcome the drawbacks of this type of out-of-context lemmatization and part-of-speech assignment. By inspecting the frequencies of stimulus and response words in the training dataset, we established a reasonable minimum frequency threshold for candidate words of 100 occurrences in UkWaC. With this threshold, only 10 response words and 16 stimulus words from the training dataset have been excluded from the vocabulary. Given the large size of the dataset, we decided that a minimal loss in coverage would be justified by the reduced computational complexity. The resulting candidate vocabulary contains 155,811 words.

Co-occurrence statistics (first-order models) Collocation data for the first-order models have been extracted from UkWaC[2] based on the vocabulary described above: both nodes (rows of the co-occurrence matrix) and collocates (columns of the co-occurrence matrix) are chosen from this vocabulary. The collection of first-order models involved the manipulation of three parameters, namely:

1. *Window size and shape*:
   - symmetric window, 2 words to the left and to the right of the node;
   - asymmetric window, 3 words to the left of the node;
   - asymmetric window, 3 words to the right of the node.
2. *Association score*:
   - co-occurrence frequency;
   - simple log-likelihood;
   - conditional probability.
3. *Index of association strength*, which determines alternative ways of quantifying the degree of association between nodes and collocates. Given two words a and b represented in a first-order model, we propose two alternative ways of quantifying the degree of association between a and b. The first option (and standard in corpus-based modeling) is to compute the *association score* between a and b. The alternative choice is based on *rank among collocates*. Given two words a and b, in our task *stimulus* and *potential response*, we consider:
   - forward rank: the rank of the potential response among the collocates of the stimulus;
   - backward rank: the rank of the stimulus among the collocates of the potential response;
   - average rank: the average of forward and backward rank.

[2]Like the window-based second-order models, collocation models have been built with UCS toolkit available at http://www.collocations.de/software.html and the `wordspace` package for R (Evert, 2014).

Bag-of-words window-based DSMs (second-order models) We employed UkWaC as a source corpus. The target words (rows) of the DSMs evaluated in this section are defined by the vocabulary described above. Evaluating the entire reference parameter set targeted in this dissertation was not feasible. Given (a) our own findings on the semantic nuances associated with different window sizes in connection with the syntagmatic vs. paradigmatic distinction and (b) our particular interest in further testing of neighbor rank in the prediction of directionality effects, we decided to restrict the scope of the evaluation to two parameters, namely window size and index of distributional relatedness. In more detail:

- *Window size*: we evaluated DSMs built from symmetric windows of 2, 4 and 16 words;
- *Index of distributional relatedness*: in parallel to the first-order setup, we compare cosine distance to the ranks among the nearest neighbors of the stimulus or response word.

As for the other parameters, our own experiments discussed earlier and in chapter 6 helped us identifying the following robust settings:

- The context words (columns) were the 50,000 most frequent context words in the respective co-occurrence matrices.
- We employed the robust *simple-log likelihood* as a feature score, in combination with a *logarithmic transformation*.
- We reduced the scored co-occurrence matrix to 1000 latent dimensions using randomized SVD (Halko et al., 2011). Note that this is the most neutral choice given our previous results.
- We adopted *cosine* as a distance metric.

8.3.2 Results

For each class of models we evaluated the different parameter values described in section 8.3.1. Table 8.12 summarizes the evaluated parameters for first-order and second-order models. Tables 8.13 and 8.14 display the results of our experiments on the training data, separately for first-order (we focus on the experiments based on a symmetric window because the asymmetric ones had a worse performance) and second-order models. Parameter configurations are reported in the *Parameter* column.[3] The number of correct responses in the lemmatized version is reported in the column *Correct*, showing how often our system predicted the correct lemma as the first candidate. Since the task of predicting exactly one word is particularly difficult, we further characterize the performance of our evaluated models by reporting the number of cases in which the correct answer from the training set was among the first 10 (< 10), 50 (< 50), or 100 (< 100) ranked candidates.

The results reported in tables 8.13 and 8.14 allowed us to identify best parameter configurations for the first-order (symmetric 2 words window, frequency, backward rank)

[3] Abbreviations used in the tables: *ass* = association score; *dist* = distance; *fwd* = forward rank; *bwd* = backward rank; *avg* = average rank.

Model	Window	Score	Relatedness Index
first-order	symmetric, 2	frequency	association score
	left 3, right 0	simple log-likelihood	forward rank
	left 0, right 3	conditional probability	backward rank
			average rank
second-order	symmetric, 2	simple log-likelihood	distance
	symmetric, 4		forward rank
	symmetric, 16		backward rank
			average rank

Table 8.12: Evaluated parameters for first- and second-order models

and second-order models (2 words window, distance). We evaluated these configurations on the test data (table 8.15).

Parameters	Correct	< 10	< 50	< 100
Freq_{ass}	2	85	372	561
Freq_{fwd}	0	77	359	550
Freq_{bwd}	**555**	973	1269	1369
Freq_{avg}	424	677	848	934
Simple-ll_{ass}	33	237	721	985
Simple-ll_{fwd}	405	760	916	947
Simple-ll_{bwd}	531	914	1141	1253
Simple-ll_{avg}	490	785	918	950
Cond.prob_{ass}	18	329	746	970
Cond.prob_{fwd}	0	77	359	550
Cond.prob_{bwd}	422	856	1129	1255
Cond.prob_{avg}	343	611	860	971

Table 8.13: First-order models - symmetric window: 2 words to the left/right of the node - training data

Parameters	Correct	< 10	< 50	< 100
2_{dist}	**264**	686	1077	1224
2_{fwd}	127	380	703	849
2_{bwd}	73	275	584	720
2_{avg}	157	436	750	911
4_{dist}	255	665	1037	1195
4_{fwd}	108	338	651	824
4_{bwd}	77	254	545	694
4_{avg}	129	397	710	862
16_{dist}	206	546	910	1062
16_{fwd}	63	252	512	667
16_{bwd}	49	188	449	581
16_{avg}	79	282	560	713

Table 8.14: Second order models – training data

Model	Correct	< 10	< 50	< 100
first-order	572 (28.6%)	1010	1303	1408
second-order	304 (15.0%)	734	1119	1256

Table 8.15: Best models (first order and second-order) – performance on test data

The results of our experiments are in line with the tendencies identified in the literature. First-order models based on direct co-occurrence (high scores are assigned to words that co-occur), outperform second-order models based on distributional similarity (smaller distances between words that occur in similar contexts). For the first-order models, the best index of association strength is the rank of the stimulus among the collocates of the potential response, which is fully congruent with the experimental setting. Surprisingly, frequency outperforms simple-log likelihood, which is usually considered to be among the best association measures for the identification of collocations. In line with the results achieved by Rapp (2014), a symmetric window of 2 words to the left and to the right of the target achieves best results.

For the second-order models, the smallest context window (2 words) achieves the best performance. Considering the good results from collocation-based models, we would have expected a better performance from larger windows, widely assumed to be more sensitive to syntagmatic relations – as confirmed in this dissertation, as well. An interesting difference between first-order and second-order models is the fact that neighbor rank works less well than the distance between vectors, while collocate rank outperforms the association scores on which it is based. This observation contrasts with what we showed in chapters 6 and 7, and in the experiments on priming datasets in this chapter: in all these cases, rank consistently outperformed distance. Among the word similarity tasks in chapter 6, however, the only case in which the use of neighbor rank did not produce significant improvements with respect to vector distance was the TOEFL multiple-choice synonymy task (in the SVD-reduced runs). Despite clear differences, the TOEFL task and the reverse association task share the property that they involve multiple stimuli. The results presented in this chapter, together with those achieved on the TOEFL task, suggest that a better strategy for the use of neighbor rank needs to be developed when multiple stimuli are involved.

An interesting research direction would be an integration of first- and second-order statistics in the process of response generation. The evaluation results reported here revealed that a very small context window achieves the best performance for second-order models: as widely acknowledged in the literature (Sahlgren, 2006) and established in the research presented in this thesis, smaller context windows highlight paradigmatic relations. First-order models, on the other hand, naturally highlight syntagmatic relations. The best second-order and first-order models from the evaluation reported in this section are likely to focus on different types of relations between response and stimulus words: this leads us to believe that an integration of the two sources may lead to improvements in performance.

8.4 Summing up

We presented two sets of studies on window-based DSMs, an intrinsic one based on a classification task derived from priming experiments, and an extrinsic one, in which the

best settings individuated in the intrinsic evaluation have been tested on the (reverse) free association test. The leading theme of this chapter is a comparison between syntagmatic and paradigmatic relations in terms of the aspects of distributional similarity that characterize them.

The results of the intrinsic evaluation show that second-order DSMs are capable of capturing both syntagmatic and paradigmatic relations, if parameters are properly tuned. Size of the co-occurrence window, as well as parameters connected to dimensionality reduction play a key role in adapting DSMs to particular relations. Even if we do not address the more specific task of distinguishing between relations (e.g., synonyms vs. antonyms; see Scheible et al. (2013) and references therein), we believe that such applications may benefit from our detailed analyses on the effects of DSM parameters.

The results of the free-association experiments reported in this chapter confirm the tendencies identified in previous studies: first-order models, based on direct co-occurrence, outperform second-order models, based on distributional similarity. We consider these experimental results a first exploration into the dynamics of the reverse association task, and we believe that our systematic evaluation of first- and second-order models represents a good starting point for future work, which targets improvements of NaDiR at many levels.

9

Conclusion

This dissertation started off by discussing the main assumption underlying Distributional Semantic Models as a method to quantify word meaning: the Distributional Hypothesis (Harris, 1954; Miller & Charles, 1991). The assumption that meaning of a word *is* its usage and can thus be operationalized in terms of the set of contexts with which this word co-occurs, is, however, not free of criticism. The main issue with the Distributional Hypothesis, and one that is at the core of the motivation of this dissertation, is its underspecification. As pointed out by Sahlgren (2008, p. 37): "The distributional hypothesis, as motivated by the works of Zellig Harris, is a strong methodological claim with a weak semantic foundation. It states that *differences* of meaning correlate with *differences* of distribution, but it neither specifies *what kind* of distributional information we should look for, nor *what kind* of meaning differences it mediates." Addressing this issue has been precisely the higher-level goal of this thesis project.

On the level of *what kind of distributional information we should look for*, we have presented the largest-scale evaluation of window-based and syntax-based DSMs, in which all possible parameter combinations are tested. Additionally, we have introduced a novel parameter, *neighbor rank*, which is cognitively motivated as it allows to model semantic similarity/relatedness as an asymmetric phenomenon.

On the level of *what kind of meaning differences are mediated by word distributions*, we have put a strong focus on cognitive tasks (along with standard word similarity tasks) and on the interpretation of the contrasts between semantic relations.

Additionally, this dissertation raises and addresses an additional issue, which had been out of the focus of previous work on DSM evaluation, namely the question of *how to assess if certain distributional properties are mediating specific meaning differences*, thus affecting DSM performance in the corresponding tasks. We have proposed a novel evaluation methodology, which is able to capture interactions between different distributional properties (the parameters of a DSM) and is robust to overfitting.

The methodological and the cognitive/semantic dimensions are the main conceptual coordinates of this work, whose contributions relate to several different domains. The main findings of this work in each of these domains are summarized below.

Neighbor rank, a cognitively-inspired parameter which has been systematically evaluated for the first time in this thesis, is a better predictor of semantic similarity/relatedness than distance in the semantic space is. Cognitively, this result stems from its capability to capture asymmetry. Mathematically, neighbor rank provides a scaling of the semantic

space which is robust to variations in density.

More specifically, our results have shown that the impact of neighbor rank is modulated by SVD: in a reduced space, the performance gain to be expected by applying rank as a relatedness index is less marked than in an unreduced space – in other words, the SVD space is more homogeneously distributed (hence the scaling effect of rank with respect to distance is less marked). The effect of rank is also modulated by the different association measures – difference between rank and distance is less marked for simple-log likelihood, t-score. Taken together with the reduced impact of rank with SVD, and with the fact that rank has a stronger effect in the (much sparser) syntax-based models with respect to the window-based ones, this result indicates in the use of neighbor rank a viable "repair parameter" to which one should definitely resort in case of very sparse spaces, or if for some reason the application of association measures is not viable.

A further advantage of an evaluation based on neighbor rank is that the DSM predictions on similarity/relatedness involve the whole vocabulary – thus producing a better estimate of the overall quality of the semantic representation – while computation of distance only involves experimental items in the evaluated datasets.

The proposed evaluation methodology revealed that parameter interactions are crucial in understanding DSM performance: not taking them into account would have prevented the identification of robust parameter settings.

In particular, the interaction between *feature score* and *feature transformation* can be considered one of the signature findings of this work, as it occurs in all experiments and with comparable best parameters. Our experiments have also shown how the manipulation of *feature score* and *feature transformation* also affects the quality of the output of the SVD reduction: the most robust unreduced configuration does not correspond to the input to most robust SVD configuration. This result is of particular relevance in the perspective of a better understanding of SVD and also of neural embeddings, which target a comparable projection of the co-occurrence facts in a lower-dimensional space. Indeed, the co-occurrence information stored in the unreduced matrix can be considered as a statistical profile of the co-occurrence facts which will also enter the computation of the embeddings, thereby providing crucial interpretation cues for their performance.

At the level of the portability of our results to new models, we have opted for a thorough exploration of the earlier stages of a DSM pipeline. Robustly negative results (i.e., lack of significant impact) for some of these (previously unexplored) parameters are extremely valuable in a field in which the potential complexity of the experimental design grows exponentially; the most notable example of negative result in this thesis is the lack of significance of parsing-specific parameters in the syntax-based experiments (parser and format of the dependencies). On the other hand, robustly identified positive trends in the earlier stages of the pipeline (i.e., co-occurrence extraction) can be employed as a departure point for further exploration in later stages of the pipeline: neighbor rank is definitely the the most promising parameter in this dissertation, along with a finer-grained manipulation of SVD.

A high-level consideration should also be made here, with regards to the relevance of the methodological contribution beyond the evaluation of count DSMs (and possibly beyond distributional modeling all together). In the deep learning era, the pace at which new embedding methods, or new hyperparameters for existing methods are being proposed is incredibly fast. In this perspective, the vision of holding up to a robust understanding of the impact of these novelties, let alone their cognitive and semantic

interpretation, may look extremely ambitious. Indeed, this would require conducting large-scale meta-studies targeting a common core of the parameter sets, while the common core is a moving target by itself. However, the theoretical desiderata and the practical guidelines defined in this thesis, along with the methodological considerations on the interpretation of the regression effects exemplified in the experimental chapters can already serve as a tool for individual researchers to get a better understanding of their own results, and to phrase that understanding in a statistical picture which is simple, reliable and easier to share with the community.

Cognitive datasets in combination with the regression methodology have helped us getting a finer-grained perspective on the parameter space. The intrinsic evaluation on the priming datasets revealed that the results obtained on similarity tasks carry over to the cognitive tasks to a reasonable extent; moreover, it contributed towards a definition of the "semantic" status of the ordering of SVD dimensions. This parameter, together with the size of the context window, is involved in the contrast between paradigmatic and syntagmatic relations. Moreover, it allowed us to characterize antonymy as a non-prototypical paradigmatic relation, one that is more topical (larger windows) and more symmetric (smaller impact of neighbor rank). The extrinsic evaluation of the best DSM settings in the reverse free association task revealed a limitation of DSMs (at least in the basic implementation of the task presented in chapter 8), as collocation models outperformed the distributional ones.

9.1 Further work on modeling reaction times in priming

A set of experiments targeting the modeling of the priming effects in the GEK dataset have already been conducted within the frame of this dissertation and have not been discussed in details in this dissertation they targeted a slightly different parameter set. Their results have been published in Lapesa & Evert (2013a,b,c). In what follows, we summarize them briefly.

These studies targeted the whole GEK (as in chapter 8) as well as its subsets (noun-noun, verb-noun, noun-verb), and zoomed in into a comparison between the relations within the datasets (agent, patient, location, etc.). Crucially, these experiments employed both directions of rank (forward and backward) as well as their average; the three alternative implementation of neighbor rank were also compared it to distance.

At the level of tasks, these studies involved a multiple choice-task as in chapter 8; a correlation task in Lapesa & Evert (2013a,b): we calculated Spearman's correlation between semantic relatedness in the distributional space and RT in the congruent condition, and employed our regression methodology to interpret DSM performance; the item-based prediction of the priming effects (difference between the congruent and incongruent condition) based on a number of second-order predictors (window-based and document-based) as well as first-order predictors Lapesa & Evert (2013c).

Taken together, the results of these experiments support of the status of *neighbor rank* as a cognitively plausible predictor:

- Neighbor rank outperformed distance, across the board.
- The best parameters for multiple-choice and correlation differ rather crucially. In particular, correlations are achieved with BNC, frequency with log or root transfor-

mation, manhattan distance, small/intermediate windows, and in the unreduced space (cf. best settings in chapter 8).

- Item-based prediction benefits from both first-order and second order predictors – besides, the directionality of the rank in the different subsets of GEK shows a very interesting pattern: forward rank is best predictor for noun-verb priming, backward rank is best predictor for noun-verb priming, average of forward and backward rank is best predictor for noun-noun priming.

- The application of the multiple-choice task to specific subsets of relations reveals patterns which are extremely interesting from the point of view of their theoretical linguistic interpretation. Note that the multiple-choice on GEK can be interpreted as a binary thematic fit task: for a specific relation, e.g., agent, high accuracy is to be interpreted as the capability of the model to tell prototypical from non prototypical agents. In this perspective we can interpret DSM performance as an answer to the question: how close are to the target verb/nouns the prototypical fillers of a specific relation? Our experiments on GEK have showed that prototypical patients (internal arguments) are closer to their head verb than instruments (indirect internal arguments) or prototypical agents (external arguments), which are in turn closer to the verb than prototypical locations (adjuncts).

9.2 Future steps

To each one of the three domains of contribution/finding listed before corresponds a set of further (and current) research directions, which are discussed below.

Neighbor rank Future research will need to target a better understanding of the effect of the different directions of rank (forward, backward) as well as different strategies for computing average rank. What is still to explore is also the hypothesis the impact on rank of target frequency, and, more in general, of the sparsity of the space. A possible way to achieve this would be to carry out an item-level analysis of the (absolute) difference between forward and backward rank. The larger such difference, the more asymmetric the relation between the two words is: this is an excellent predicted value for a regression analysis, with relevant word-level features (e.g., frequency, part-of-speech, number of non-zero entries) as well as DSM-related features (e.g., parameters values) as predictors. Last, the hypothesis that the density of the semantic representation reduces its asymmetry will find its natural next test in the application of neighbor rank in neural word embeddings.

As a preliminary step for further research on asymmetry in distributional semantics, we have already collected a dataset in which directionality of the judgment is taken into account as an experimental variable (Lapesa, Schulte im Walde, & Evert, 2014). Focussing on paradigmatically related pairs, and after a careful stratified sampling of the items, we and asked the (Amazon Mechanical Turk) subjects to rate the degree of relatedness between two words (e.g., *artist* and *painter*) with respect to a specific relation (e.g., *synonymy*), and presented the pair in both orders (e.g., *artist-synonym-painter* vs. *painter-synonym-artist*). The question at issue is to what extent asymmetry would affect the targeted relations (e.g., synonymy, antonymy, hyponymy) with regard to different parts of speech (verbs vs. adjectives vs. nouns). We expect the comparison

across parts of speech to highlight conceptual differences of the three relations across word classes: for example, the concept of hyponymy has been widely investigated with respect to nouns, but little attention has been devoted to its application to verbs, and even less to adjectives

Regression analysis As already pointed out throughout the thesis, the analysis of higher-level interactions is the necessary further step in the refinement of the proposed regression methodology.

Given the high number of data points, we have so far focussed only two-way interactions, but the clear next step towards a more powerful analysis would be to a) get rid of weak parameters values across the board (e.g., manhattan, sigmoid) and introduce three-way interactions; b) introduce the datasets as a predictor of DSM performance – this would make the comparisons among datasets more straightforward; c) align the parameters of the different model classes and train a larger regression model – note that many of the parameters in this study are already aligned, and many of the remaining ones can straightforwardly be aligned: window size in window-based with path length in syntax-based; dependency-filtered are the syntax-based version of undirected window, while dependency-typed can is the syntax-based counterpart of a directed window encoding relative position of the feature with respect to the target word.

Cognitive modeling The experiments on cognitive datasets described in chapter 8 represent the natural starting point for two research lines, which cross each other in the strive for an integration of first-order (collocation) and second order (distributional) models.

As far as the modeling of priming data is concerned, a more exhaustive distributional account of the experimental effects requires the actual comparison between reaction times and distributional similarities. In chapter 3 we already discussed what the problematic points of such an approach would be; in section 9.1 we have summarized the results of further correlation and item-based modeling of reaction times in GEK; these studies can be considered as extremely promising pilot for future work, involving also SPP.

As far as the reverse association task is concerned, even if we have established that first-order co-occurrence models outperform DSMs when predicting free associations, there is still a possibility for the two classes of models to be (at least in part) complementary in their predictions. This could be the task for an oracle, or, even better, for an ensemble model trained in the task of finding the appropriate way of combining first order and second order information for the responses, based on a number of item-level predictors (e.g., frequency, part of speech, or finer grained semantic features such as concreteness, valence, etc.). In this connection, we created a new free-association dataset larger than the CogALex one, merging EAT and USF and carefully sampling the items according to frequency (to avoid, or at least alleviate frequency bias). The dataset and the results of preliminary experiments have been presented by Evert & Lapesa (2017); further modeling experiments involving neural embeddings and the integration of different sources of corpus-based information have been performed, setting the stage for an exhaustive comparison of such sources in what still remains an extremely challenging cognitive task.

A

Clustering implementation: `pam` vs. CLUTO

This section summarizes the results of the comparison between the clustering performance achieved by the `pam` function from the `cluster` R package (with standard settings) and the performance achieved by the CLUTO toolkit (Karypis, 2003).

Table A.1 reports the results of this comparison, on the unreduced and reduced experimental runs.

As CLUTO relies on cosine to perform clustering, we present the comparison between PAM+cosine and CLUTO. We compare performance of PAM achieved with the two indexes of distributional relatedness (*distance* and *rank*) to the performance of CLUTO, separately for reduced and unreduced runs.

We conducted paired t-tests to check for significant differences between `pam` and CLUTO in our clustering experiments: table A.1 reports, for every comparison, the difference of means (`pam` minus CLUTO) and the significance value. For example, the unreduced/distance/AP cell from table tells us that, for the unreduced runs and the Almuhareb-Poesio dataset, `pam` was slightly but significantly better CLUTO. The cases in which `pam` turned out to be better - or at least not worse - than CLUTO are highlighted in bold.

Dataset	Unreduced				Reduced			
	Distance		Rank		Distance		Rank	
dataset	diff.means	p	diff.means	p	diff.means	p	diff.means	p
AP	**0.01050**	***	**0.04590**	***	**0.02814**	***	**0.03918**	***
BATTIG	-0.00010		**0.05588**	***	**0.02310**	***	**0.03979**	***
ESSLLI	**0.01050**	***	**0.04590**	***	-0.00740	***	**0.00066**	***
MITCHELL	-0.01683	***	**0.07280**	***	**0.03999**	***	**0.05249**	***

Table A.1: Comparison between `pam` and CLUTO

B

Best Models

B.1 Window-based models

TOEFL

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
ukwac	2	undirected	20000	nnzero	MI	none	cosine	rank	87.50
ukwac	2	undirected	5000	f	simple-ll	log	cosine	dist	87.50
ukwac	4	undirected	20000	f	simple-ll	none	man	rank	87.50
ukwac	4	undirected	20000	nnzero	simple-ll	none	man	rank	87.50

Table B.1: TOEFL, unreduced, best models – 6 runs tied for best result (4 hand-picked examples shown)

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
ukwac	2	undirected	f	5000	MI	none	cosine	rank	900	100	98.75
ukwac	4	directed	f	50000	t-score	log	cosine	rank	900	100	98.75
ukwac	4	undirected	f	50000	t-score	root	cosine	dist	900	100	98.75
ukwac	4	directed	f	5000	simple-ll	log	cosine	dist	900	100	98.75

Table B.2: TOEFL, reduced, best models – 23 models tied for best result (4 hand-picked examples shown)

Ratings

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	r
wacky	4	undirected	100000	f	simple-ll	none	man	rank	0.88
wacky	4	undirected	100000	nnzero	simple-ll	none	man	rank	0.88
wacky	4	undirected	100000	f	simple-ll	root	cosine	rank	0.87
wacky	4	undirected	100000	f	z-score	none	cosine	rank	0.86

Table B.3: RG65 dataset, unreduced, best models – 2 models tied for best r, 2 additional hand-picked models with similar performance are shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	r
ukwac	16	undirected	nnzero	20000	MI	none	cosine	rank	700	100	0.89
ukwac	8	directed	f	20000	MI	none	cosine	rank	700	100	0.89
wacky	4	directed	nnzero	50000	simple-ll	log	cosine	rank	700	50	0.89
wacky	4	undirected	f	100000	z-score	log	cosine	rank	900	50	0.89

Table B.4: RG65 dataset, reduced, best models – 19 models tied for best result (4 hand-picked examples shown)

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	r
wacky	16	undirected	100000	f	z-score	none	cosine	rank	0.73
wacky	16	undirected	100000	nnzero	z-score	none	cosine	rank	0.73
ukwac	16	undirected	100000	nnzero	z-score	none	cosine	rank	0.70
wacky	16	directed	100000	f	simple-ll	root	cosine	rank	0.71

Table B.5: WS353 dataset, unreduced, best models – 2 models tied for best r, 2 additional hand-picked models with similar performance shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	r
wacky	16	directed	f	5000	MI	none	man	rank	900	50	0.73
wacky	16	undirected	f	5000	MI	none	man	rank	900	50	0.72
wacky	16	undirected	f	5000	z-score	log	man	rank	900	50	0.72
wacky	16	directed	f	10000	z-score	root	man	rank	900	50	0.72

Table B.6: WS353 dataset, reduced, best model – 3 additional hand-picked models with similar performance are shown

Clustering

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	purity
wacky	1	directed	f	50000	MI	none	cosine	rank	0.73
wacky	1	directed	f	50000	z-score	log	cosine	rank	0.73
wacky	1	undirected	f	10000	z-score	log	cosine	rank	0.73
wacky	1	undirected	f	100000	simple-ll	log	cosine	rank	0.73

Table B.7: AP dataset, unreduced, best models – 7 models tied for best result (4 hand-picked examples shown)

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	purity
ukwac	4	directed	nnzero	10000	t-score	log	man	rank	900	50	0.76
wacky	1	directed	nnzero	10000	z-score	log	man	rank	900	50	0.75
wacky	1	undirected	f	20000	simple-ll	log	man	rank	900	50	0.75
wacky	2	directed	f	100000	z-score	log	cosine	rank	500	0	0.75

Table B.8: AP dataset, reduced – best model (plus 3 additional hand-picked models)

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	purity
ukwac	16	directed	nnzero	5000	simple-ll	none	man	dist	0.99
wacky	4	undirected	f	100000	MI	none	cosine	rank	0.99

Table B.9: BATTIG dataset, unreduced – best models

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	purity
ukwac	1	undirected	f	20000	Dice	root	man	rank	300	100	0.99
ukwac	2	undirected	f	100000	freq	log	cosine	dist	300	50	0.99
wacky	16	undirected	f	50000	z-score	log	man	dist	500	50	0.99
wacky	8	undirected	f	10000	Dice	root	man	rank	500	0	0.99

Table B.10: BATTIG dataset, reduced, best models – 1037 models tied for best result (4 hand-picked examples shown)

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	purity
wacky	1	undirected	50000	f	z-score	none	man	dist	0.93
wacky	1	undirected	50000	nnzero	z-score	none	man	dist	0.93
wacky	2	directed	100000	f	z-score	none	man	dist	0.93
wacky	2	directed	100000	nnzero	z-score	none	man	dist	0.93

Table B.11: ESSLLI dataset, unreduced, best models – 5 models tied for best results, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	purity
wacky	16	directed	nnzero	50000	z-score	none	man	dist	900	0	0.98
ukwac	1	directed	nnzero	100000	simple-ll	log	cosine	dist	100	50	0.95
ukwac	2	undirected	f	50000	tf.idf	none	man	dist	700	0	0.95
wacky	8	undirected	f	100000	tf.idf	root	man	rank	500	0	0.95

Table B.12: ESSLLI dataset, reduced – best model (plus 3 additional hand-picked models)

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	purity
bnc	1	undirected	100000	f	z-score	log	cosine	rank	0.97
bnc	2	undirected	10000	nnzero	z-score	root	cosine	rank	0.97
bnc	4	undirected	100000	f	z-score	none	man	rank	0.97
bnc	4	undirected	50000	f	z-score	none	man	rank	0.97

Table B.13: MITCHELL dataset, unreduced, best models – 6 models tied for best results, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	purity
bnc	2	undirected	nnzero	100000	simple-ll	log	cosine	rank	900	0	0.97
bnc	2	undirected	f	50000	simple-ll	log	cosine	rank	700	0	0.97
bnc	2	undirected	nnzero	50000	simple-ll	log	cosine	rank	900	0	0.97

Table B.14: MITCHELL dataset, reduced, best models – 3 models tied for best result

Semantic Priming

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
ukwac	2	directed	5000	nnzero	z-score	root	cosine	rank	98.39
ukwac	2	directed	10000	f	simple-ll	log	cosine	rank	98.17
wacky	2	directed	20000	nnzero	z-score	log	cosine	rank	98.17
wacky	2	directed	20000	nnzero	z-score	root	cosine	rank	98.17

Table B.15: SYN, unreduced, best model – (plus 3 additional hand-picked models)

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
ukwac	2	directed	100000	nnzero	MI	root	cosine	rank	100	900	99.08
ukwac	8	undirected	20000	f	MI	root	cosine	dist	100	900	99.08
ukwac	8	undirected	20000	f	MI	root	cosine	rank	100	900	99.08

Table B.16: SYN, reduced, best models

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
wacky	8	directed	5000	nnzero	z-score	root	cosine	rank	100.00
ukwac	8	directed	10000	f	z-score	none	man	dist	100.00
wacky	16	undirected	5000	f	simple-ll	log	man	rank	100.00
wacky	8	directed	100000	nnzero	simple-ll	root	cosine	dist	100.00

Table B.17: ANT, unreduced, best models – 3261 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
wacky	16	undirected	20000	f	tf.idf	root	cosine	dist	100	700	100.00
wacky	2	directed	50000	f	z-score	root	cosine	dist	0	100	100.00
ukwac	16	directed	10000	nnzero	MI	none	man	rank	50	700	100.00
ukwac	4	undirected	5000	f	tf.idf	root	man	rank	0	500	100.00

Table B.18: ANT, reduced, best models – 15345 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
wacky	2	undirected	100000	f	t-score	sigmoid	cosine	rank	100.00
ukwac	1	undirected	50000	f	MI	log	cosine	rank	100.00
ukwac	1	directed	20000	nnzero	t-score	root	cosine	rank	100.00
ukwac	1	undirected	10000	nnzero	tf.idf	root	cosine	rank	100.00

Table B.19: COH, unreduced, best models – 1067 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
wacky	2	directed	50000	nnzero	frequency	root	cosine	rank	0	500	100.00
wacky	2	undirected	20000	nnzero	MI	none	cosine	rank	0	300	100.00
wacky	1	undirected	50000	nnzero	MI	log	cosine	rank	0	500	100.00
ukwac	1	undirected	50000	f	Dice	root	man	rank	100	900	100.00

Table B.20: COH, reduced, best models – 9804 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
ukwac	4	undirected	10000	f	z-score	none	cosine	rank	97.92
ukwac	4	directed	100000	f	z-score	none	cosine	rank	97.92
ukwac	4	directed	100000	nnzero	z-score	none	cosine	rank	97.92
ukwac	4	undirected	100000	nnzero	z-score	none	cosine	rank	97.92

Table B.21: FPA, unreduced, best models – 10 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
ukwac	16	directed	20000	f	z-score	root	cosine	rank	100	900	98.61
ukwac	16	undirected	20000	nnzero	z-score	root	cosine	rank	100	900	97.92
ukwac	16	undirected	20000	nnzero	z-score	root	cosine	dist	100	900	97.92
ukwac	8	undirected	50000	nnzero	z-score	none	cosine	rank	0	500	97.92

Table B.22: FPA, reduced, best model – 3 additional hand-picked models with similar performance are shown

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
ukwac	4	directed	10000	nnzero	z-score	none	cosine	rank	97.75
ukwac	8	undirected	100000	f	z-score	none	cosine	rank	97.75
ukwac	8	undirected	100000	f	z-score	none	cosine	dist	97.75
ukwac	8	directed	50000	f	z-score	none	cosine	rank	97.75

Table B.23: BPA, unreduced, best models – 33 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
ukwac	16	undirected	100000	f	MI	none	cosine	dist	50	700	98.88
ukwac	4	undirected	50000	f	MI	root	cosine	rank	50	700	98.88
ukwac	4	undirected	100000	f	z-score	log	cosine	dist	50	500	98.88
ukwac	8	directed	50000	nnzero	z-score	log	cosine	rank	50	300	98.88

Table B.24: BPA, reduced, best models – 181 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	c.dim	criterion	score	transf	metric	rel.ind	acc
ukwac	16	undirected	20000	nnzero	z-score	none	cosine	rank	95.30
wacky	16	directed	100000	f	z-score	none	cosine	rank	95.30
ukwac	16	directed	100000	nnzero	z-score	none	cosine	rank	95.30
ukwac	16	undirected	20000	f	z-score	none	cosine	rank	95.30

Table B.25: GEK, unreduced, best models – 6 runs tied for best result, 4 hand-picked examples shown

corpus	window	direction	criterion	c.dim	score	transf	metric	rel.ind	red.dim	dim.skip	acc
ukwac	16	directed	10000	f	z-score	log	man	rank	100	900	97.03
ukwac	8	undirected	50000	f	simple-ll	log	cosine	rank	50	700	96.78
ukwac	8	undirected	10000	nnzero	simple-ll	log	man	rank	100	900	96.78
ukwac	4	undirected	20000	nnzero	t-score	none	man	rank	50	700	96.78

Table B.26: GEK, reduced, best model – 3 additional hand-picked models with similar performance are shown

B.2 Syntax-based models

TOEFL

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	wacky	stanford	core	ccproc	1	50k	z-score	none	cosine	rank	85.00
typed	wacky	stanford	core	basic	2	100k	z-score	none	cosine	rank	83.75
typed	wacky	stanford	core	basic	2	50k	z-score	none	cosine	rank	83.75

Table B.2.27: TOEFL, unreduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	ukwac	malt	ext	basic	2	5k	t-score	log	500	100	cosine	rank	93.75
filtered	ukwac	stanford	ext	basic	3	5k	t-score	log	500	900	cosine	rank	93.75
filtered	wacky	stanford	ext	basic	2	10k	simple-ll	log	500	900	cosine	rank	93.75
typed	ukwac	stanford	ext	basic	4	50k	MI	none	100	700	cosine	dist	91.25
typed	ukwac	stanford	ext	basic	4	50k	MI	none	100	900	cosine	dist	91.25
typed	ukwac	stanford	ext	basic	1	100k	MI	root	100	900	cosine	dist	91.25

Table B.2.28: TOEFL, reduced. Filtered (3 runs tied for best result) vs. Typed (3 runs tied for best result)

Ratings

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	r
filtered	wacky	malt	ext	ccproc	1	50k	MI	none	cosine	rank	0.88
typed	wacky	malt	core	ccproc	1	100k	z-score	none	man	rank	0.80

Table B.2.29: RG65, unreduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	r
filtered	ukwac	malt	core	basic	4	50k	MI	none	50	500	cosine	rank	0.88
typed	wacky	malt	core	basic	1	100k	z-score	log	100	900	cosine	rank	0.87

Table B.2.30: RG65, reduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	r
filtered	ukwac	stanford	ext	ccproc	4	50k	z-score	none	cosine	rank	0.71
typed	ukwac	stanford	ext	basic	1	100k	z-score	root	cosine	rank	0.59

Table B.2.31: WS353, unreduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	r
filtered	ukwac	stanford	core	ccproc	3	100k	z-score	root	50	900	cosine	rank	0.72
typed	ukwac	stanford	ext	basic	1	100k	MI	none	50	900	cosine	rank	0.66

Table B.2.32: WS353, reduced, best models - Filtered vs. Typed

Clustering

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	purity
filtered	ukwac	malt	ext	basic	1	100k	z-score	log	cosine	rank	0.75
typed	wacky	stanford	ext	ccproc	1	100k	z-score	none	man	rank	0.75
typed	wacky	malt	ext	ccproc	1	100k	z-score	root	cosine	rank	0.75
typed	wacky	stanford	ext	ccproc	1	100k	z-score	none	man	rank	0.75

Table B.2.33: AP, unreduced, best models - Filtered vs. Typed (3 runs tied for best result)

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	purity
filtered	wacky	malt	core	ccproc	1	20k	t-score	none	0	900	man	rank	0.75
typed	ukwac	stanford	ext	basic	1	100k	z-score	root	0	300	cosine	rank	0.78

Table B.2.34: AP, reduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	purity
filtered	bnc	malt	ext	basic	2	100k	z-score	none	man	rank	0.98
filtered	ukwac	malt	core	basic	2	100k	simple-ll	log	cosine	rank	0.98
filtered	wacky	stanford	ext	basic	2	50k	z-score	none	man	dist	0.98
typed	ukwac	stanford	core	ccproc	1	100k	Dice	root	cosine	rank	0.95

Table B.2.35: BATTIG, unreduced, best models - Filtered (46 runs tied for best result, 3 hand-picked examples shown) vs.Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	purity
filtered	bnc	malt	core	basic	4	50k	z-score	root	0	500	cosine	rank	0.99
filtered	ukwac	malt	core	ccproc	4	100k	z-score	none	100	500	man	rank	0.99
filtered	ukwac	malt	ext	basic	1	100k	freq	log	50	300	cosine	dist	0.99
typed	ukwac	stanford	core	ccproc	1	100k	z-score	root	50	100	cosine	rank	1.00

Table B.2.36: BATTIG, reduced, best models - Filtered (520 runs tied for best result, 3 hand-picked examples shown) vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	purity
filtered	bnc	stanford	ext	basic	1	50k	MI	none	cosine	rank	0.91
filtered	wacky	stanford	ext	basic	1	100k	simple-ll	log	man	rank	0.91
filtered	ukwac	stanford	ext	basic	1	50k	z-score	log	man	rank	0.91
filtered	wacky	stanford	ext	ccproc	1	100k	z-score	none	man	dist	0.91
typed	bnc	malt	ext	basic	1	20k	t-score	sigmoid	cosine	rank	0.89
typed	bnc	malt	ext	basic	1	50k	MI	root	cosine	rank	0.89

Table B.2.37: ESSLLI, unreduced, best models - Filtered (4 runs tied for best result) vs. Typed (2 runs tied for best result)

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	purity
filtered	ukwac	stanford	core	basic	1	100k	simple-ll	log	50	700	cosine	dist	0.98
filtered	ukwac	stanford	core	basic	1	50k	tf.idf	root	50	500	cosine	dist	0.98
filtered	wacky	malt	ext	basic	3	100k	z-score	none	0	700	man	rank	0.98
typed	ukwac	stanford	ext	basic	1	100k	simple-ll	log	50	100	cosine	rank	0.98

Table B.2.38: ESSLLI, reduced, best models - Filtered (29 runs tied for best result 3 hand-picked examples shown) vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	purity
filtered	bnc	ext	malt	basic	1	100k	simple-ll	log	cosine	rank	0.93
filtered	ukwac	ext	malt	basic	3	10k	simple-ll	root	cosine	rank	0.93
filtered	bnc	ext	stanford	basic	2	50k	simple-ll	root	cosine	rank	0.93
typed	bnc	ext	stanford	basic	1	100k	z-score	none	man	dist	0.90
typed	bnc	ext	stanford	basic	1	50k	z-score	none	man	dist	0.90
typed	bnc	ext	stanford	basic	2	100k	z-score	none	cosine	rank	0.90

Table B.2.39: MITCHELL, unreduced, best models - Filtered (33 runs tied for best result, 3 hand-picked examples shown) vs. Typed (3 runs tied for best result).

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	purity
filtered	bnc	stanford	ext	basic	2	20k	z-score	root	0	700	cosine	rank	0.97
typed	bnc	malt	ext	ccproc	1	100k	Dice	root	50	100	cosine	rank	0.95
typed	bnc	stanford	ext	basic	1	100k	z-score	root	50	300	cosine	rank	0.95

Table B.2.40: MITCHELL, reduced, best models - Filtered vs. Typed (2 runs tied for best result)

Semantic Priming

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	wacky	stanford	ext	ccproc	1	100k	z-score	log	cosine	rank	97.94
filtered	wacky	stanford	ext	ccproc	1	50k	simple-ll	log	cosine	rank	97.94
filtered	wacky	stanford	ext	ccproc	1	50k	z-score	root	cosine	rank	97.94
filtered	wacky	stanford	ext	ccproc	1	5k	z-score	none	man	rank	97.94
typed	wacky	stanford	ext	ccproc	1	100k	z-score	none	cosine	rank	96.56
typed	wacky	stanford	ext	basic	2	100k	z-score	none	cosine	rank	96.56

Table B.2.41: SYN, unreduced, best models - Filtered (4 runs tied for best result) vs. Typed (2 runs tied for best result)

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	ukwac	stanford	ext	ccproc	1	50k	z-score	log	100	900	cosine	rank	99.31
typed	ukwac	stanford	ext	basic	1	100k	tf.idf	log	50	900	cosine	rank	97.25

Table B.2.42: SYN, reduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	bnc	malt	core	basic	4	10k	frequency	none	man	rank	100.00
filtered	ukwac	stanford	ext	basic	3	100k	z-score	none	cosine	rank	100.00
filtered	wacky	stanford	core	basic	3	20k	t-score	none	cosine	rank	100.00
typed	bnc	malt	core	ccproc	1	50k	MI	none	cosine	rank	100.00
typed	ukwac	malt	ext	ccproc	1	10k	Dice	none	cosine	rank	100.00
typed	wacky	malt	core	basic	4	50k	simple-ll	log	man	rank	100.00

Table B.2.43: ANT, unreduced, best models - Filtered (5387 runs tied for best result, 3 hand-picked examples shown) vs. Typed (1469 runs tied for best result, 3 hand-picked examples shown).

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	bnc	malt	core	ccproc	4	50k	MI	none	0	700	cosine	rank	100.00
filtered	wacky	stanford	ext	ccproc	2	100k	MI	log	0	900	cosine	rank	100.00
filtered	ukwac	stanford	ext	ccproc	1	5k	MI	root	0	300	cosine	dist	100.00
typed	ukwac	malt	core	basic	4	20k	tf.idf	log	100	700	cosine	dist	100.00
typed	wacky	stanford	ext	ccproc	3	5k	simple-ll	sigmoid	50	500	man	rank	100.00
typed	wacky	stanford	ext	ccproc	1	5k	t-score	cosine	50	900	cosine	rank	100.00

Table B.2.44: ANT, reduced, best models - Filtered (23209 runs tied for best result, 3 hand-picked examples shown) vs. Typed: (805 runs tied for best result, 3 hand-picked examples shown).

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	wacky	stanford	ext	ccproc	1	50k	simple-ll	root	cosine	rank	100.00
filtered	ukwac	stanford	ext	basic	1	50k	Dice	none	cosine	rank	100.00
filtered	bnc	stanford	core	basic	1	100k	simple-ll	log	cosine	rank	100.00
typed	wacky	stanford	ext	ccproc	1	10k	simple-ll	none	man	dist	100.00
typed	wacky	stanford	ext	ccproc	2	50k	z-score	none	cosine	rank	100.00
typed	ukwac	stanford	core	ccproc	1	100k	MI	sigmoid	cosine	rank	100.00

Table B.2.45: COH, unreduced, best models - Filtered (1139 runs tied for best result, 3 hand-picked examples shown) vs. Typed (721 runs tied for best result, 3 hand-picked examples shown).

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	ukwac	stanford	ext	basic	2	100k	MI	log	50	500	cosine	rank	100.00
filtered	wacky	stanford	ext	ccproc	2	5k	z-score	log	50	900	cosine	rank	100.00
filtered	ukwac	malt	core	basic	1	10k	MI	none	50	700	man	rank	100.00
typed	bnc	stanford	core	ccproc	1	10k	MI	root	0	900	cosine	rank	100.00
typed	wacky	stanford	ext	ccproc	3	50k	z-score	root	0	700	cosine	rank	100.00
typed	ukwac	stanford	ext	ccproc	1	100k	t-score	sigmoid	0	700	cosine	rank	100.00

Table B.2.46: COH, reduced, best models - Filtered (8237 runs tied for best result, 3 hand-picked examples shown) vs. Typed (2617 runs tied for best result, 3 hand-picked examples shown).

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	ukwac	stanford	core	basic	2	100k	Dice	none	cosine	rank	97.22
filtered	ukwac	stanford	core	basic	2	50k	z-score	none	cosine	rank	97.22
filtered	wacky	malt	ext	basic	4	50k	z-score	none	cosine	rank	97.22
typed	wacky	stanford	core	basic	1	50k	MI	none	cosine	rank	88.19
typed	wacky	stanford	ext	basic	1	100k	MI	none	cosine	rank	88.19
typed	wacky	stanford	ext	basic	1	50k	z-score	none	man	rank	88.19

Table B.2.47: FPA, unreduced, best models - Filtered (9 runs tied for best result, 3 hand-picked examples shown). Typed (5 runs tied for best result, 3 hand-picked examples shown)

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	wacky	malt	ext	ccproc	3	20k	z-score	none	0	900	cosine	dist	97.22
filtered	ukwac	stanford	core	ccproc	4	10k	tf.idf	none	50	50	man	rank	97.22
filtered	wacky	stanford	ext	basic	3	10k	z-score	none	0	900	cosine	rank	97.22
typed	ukwac	stanford	ext	basic	1	100k	z-score	root	50	900	cosine	rank	90.28

Table B.2.48: FPA, reduced, best models - Filtered (15 runs tied for best result, 3 hand-picked examples shown) vs. Typed.

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	ukwac	malt	core	basic	3	100k	z-score	root	cosine	rank	97.75
filtered	ukwac	malt	ext	basic	2	100k	Dice	log	cosine	rank	97.75
filtered	ukwac	malt	ext	ccproc	4	20k	z-score	none	cosine	rank	97.75
typed	ukwac	malt	ext	basic	1	100k	z-score	root	cosine	rank	92.13

Table B.2.49: BPA, unreduced, best models - Filtered (43 runs tied for best result, 3 hand-picked examples shown) vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	bnc	malt	ext	ccproc	3	100k	MI	none	50	700	cosine	rank	98.88
filtered	ukwac	malt	core	basic	4	100k	simple-ll	log	50	700	cosine	rank	98.88
filtered	ukwac	malt	core	ccproc	3	100k	Dice	root	50	500	cosine	rank	98.88
typed	ukwac	stanford	core	basic	2	100k	Dice	root	100	900	cosine	rank	95.51
typed	ukwac	stanford	core	basic	2	100k	Dice	root	100	900	cosine	rank	95.51

Table B.2.50: BPA, reduced, best models - Filtered (365 runs tied for best result) vs. Typed (2 models tied for best result)

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	metric	rel.ind	acc
filtered	ukwac	stanford	ext	basic	4	50k	z-score	none	cosine	rank	95.54
typed	ukwac	stanford	ext	basic	1	100k	z-score	none	cosine	rank	87.13

Table B.2.51: GEK, unreduced, best models - Filtered vs. Typed

	corpus	parser	d.group	d.style	p.len	c.dim	score	transf	d.skip	n.dim	metric	rel.ind	acc
filtered	ukwac	malt	ext	basic	4	10k	MI	none	50	700	man	rank	95.79
filtered	ukwac	malt	ext	basic	4	50k	Dice	root	50	900	cosine	dist	95.79
filtered	ukwac	malt	ext	basic	4	50k	Dice	root	50	900	cosine	rank	95.79
filtered	ukwac	stanford	ext	ccproc	3	50k	Dice	root	50	900	cosine	dist	95.79
typed	ukwac	malt	core	basic	2	100k	tf.idf	log	50	300	cosine	dist	89.60
typed	ukwac	malt	ext	basic	2	100k	frequency	log	50	900	cosine	dist	89.60

Table B.2.52: GEK, reduced, best models - Filtered (4 runs tied for best result). Typed (2 runs tied for best result).

C

Distribution of Performance

C.1 TOEFL

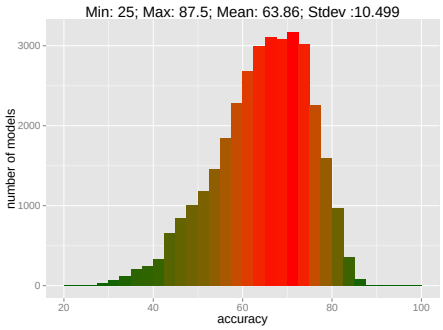

Figure C.1.1: Win.based, unreduced

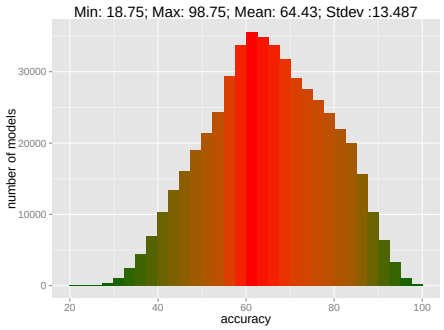

Figure C.1.2: Win.based, reduced

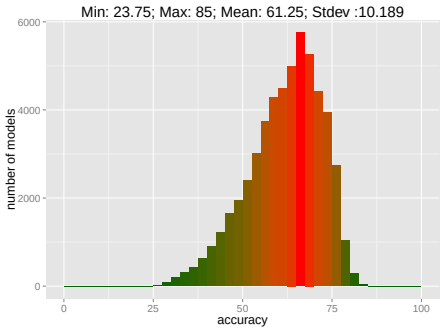

Figure C.1.3: Dep.filtered, unreduced

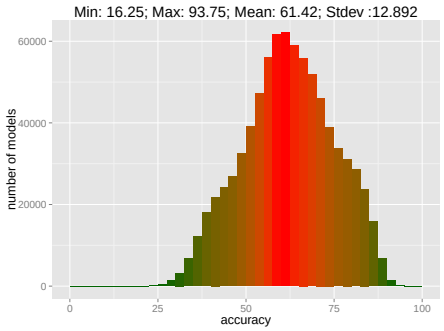

Figure C.1.4: Dep.filtered, reduced

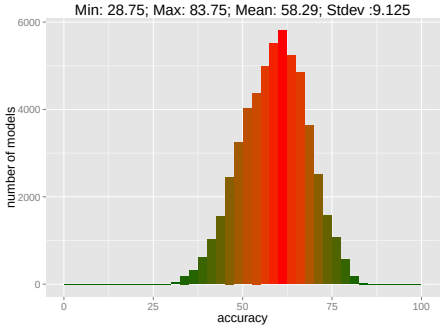

Figure C.1.5: Dep.typed, unreduced

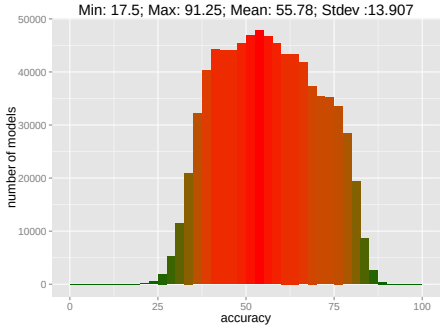

Figure C.1.6: Dep.typed, reduced

C.2 WS353

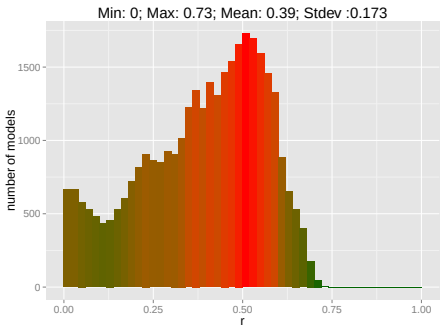

Figure C.2.7: Win.based, unreduced

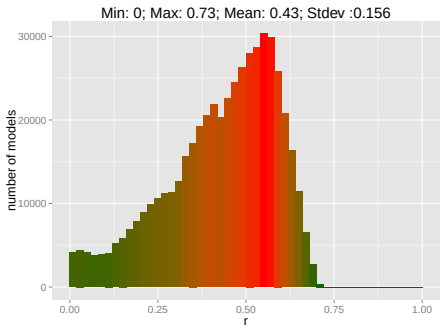

Figure C.2.8: Win.based, reduced

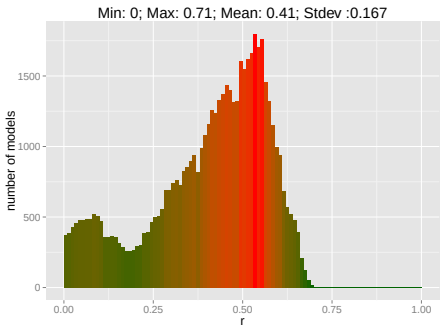

Figure C.2.9: Dep.filtered, unreduced

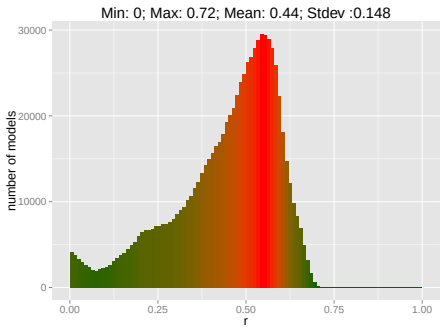

Figure C.2.10: Dep.filtered, reduced

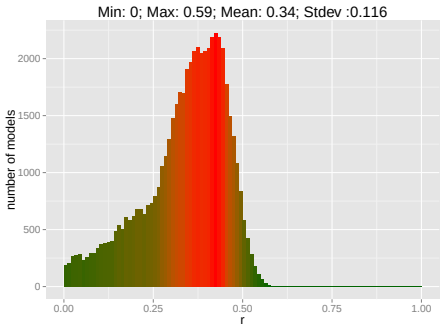

Figure C.2.11: Dep.typed, unreduced

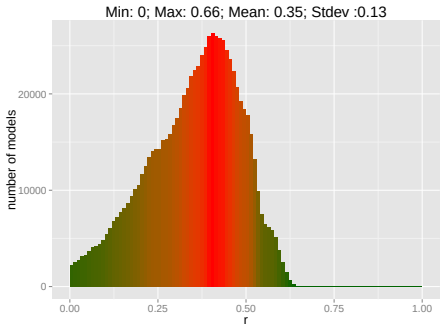

Figure C.2.12: Dep.typed, reduced

C.3 RG65

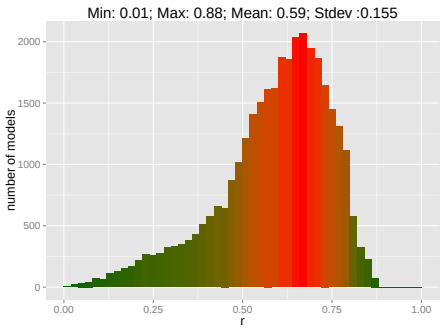

Figure C.3.13: Win.based, unreduced

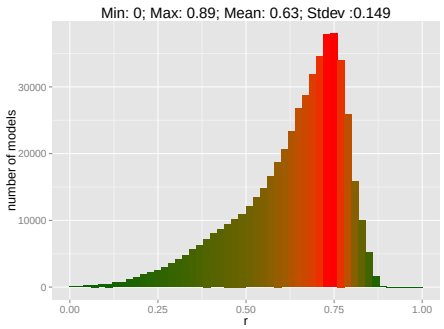

Figure C.3.14: Win.based, reduced

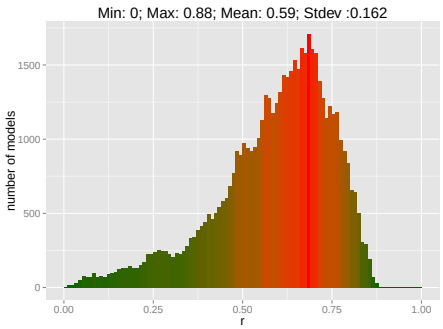

Figure C.3.15: Dep.filtered, unreduced

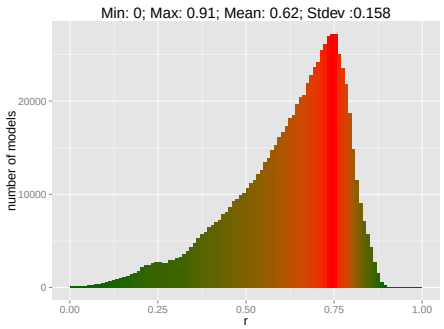

Figure C.3.16: Dep.filtered, reduced

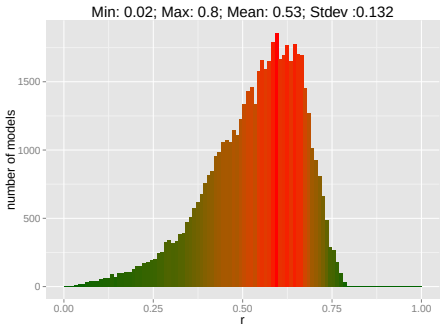

Figure C.3.17: Dep.typed, unreduced

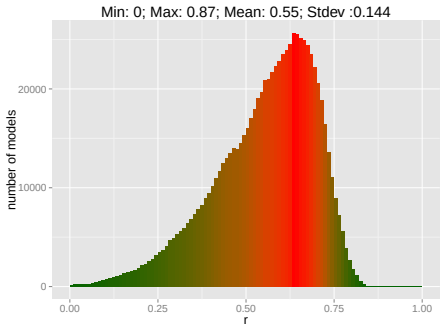

Figure C.3.18: Dep.typed, reduced

C.4 AP

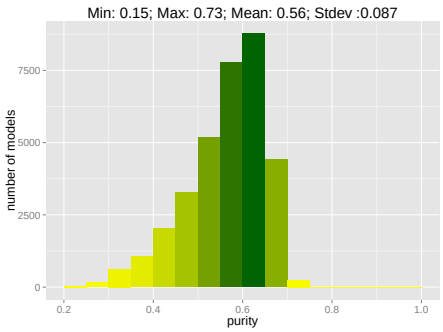

Figure C.4.19: Win.based, unreduced

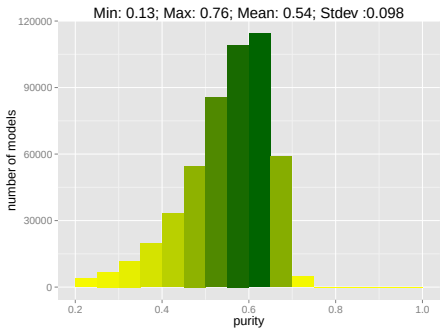

Figure C.4.20: Win.based, reduced

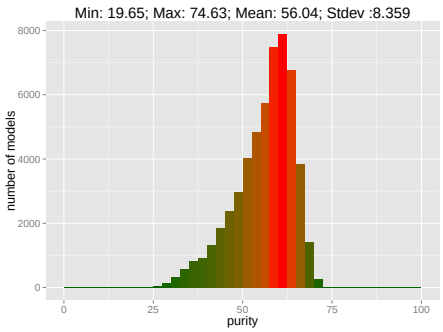

Figure C.4.21: Dep.filtered, unreduced

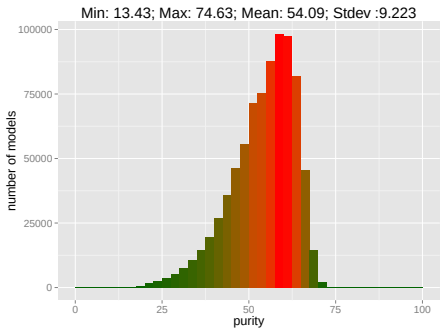

Figure C.4.22: Dep.filtered, reduced

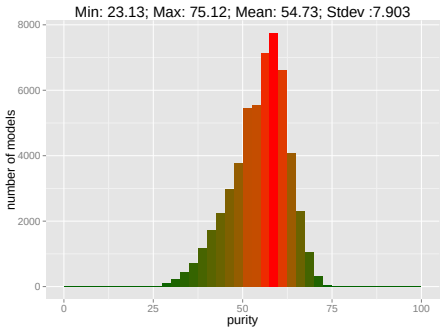

Figure C.4.23: Dep.typed, unreduced

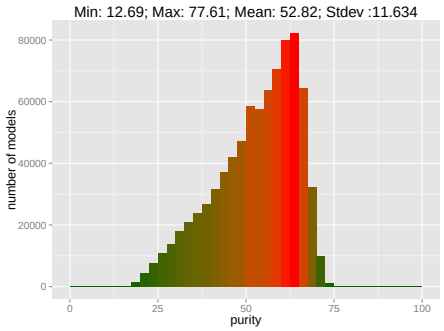

Figure C.4.24: Dep.typed, reduced

C.5 BATTIG

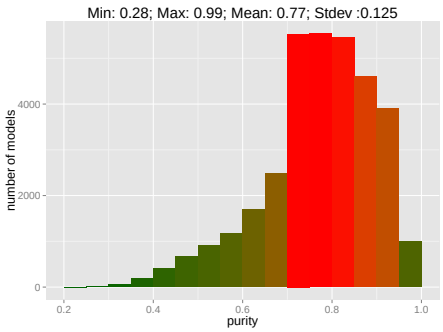

Figure C.5.25: Win.based, unreduced

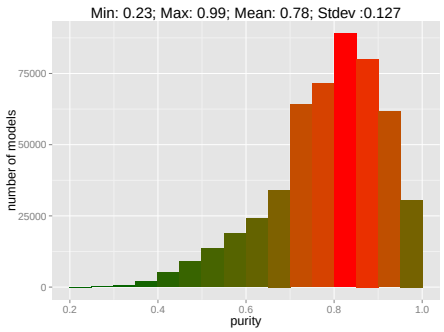

Figure C.5.26: Win.based, reduced

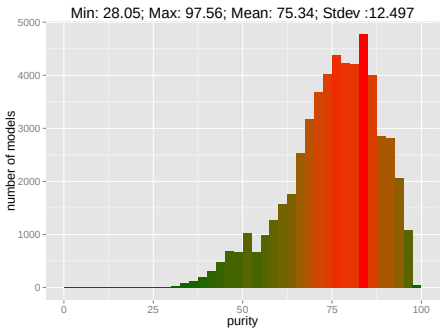

Figure C.5.27: Dep.filtered, unreduced

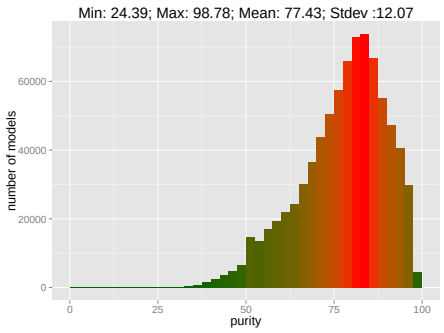

Figure C.5.28: Dep.filtered, reduced

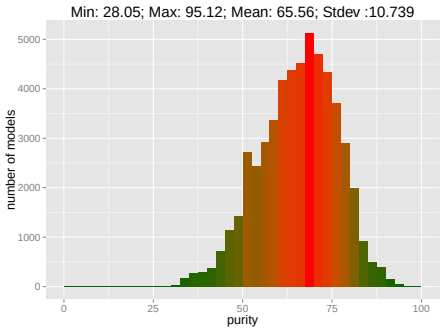

Figure C.5.29: Dep.typed, unreduced

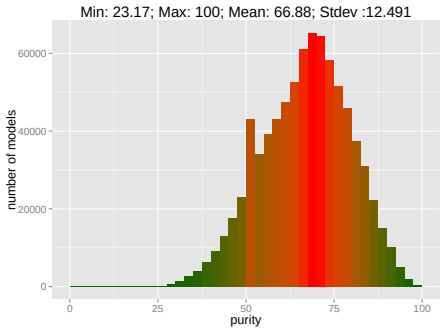

Figure C.5.30: Dep.typed, reduced

C.6 ESSLLI

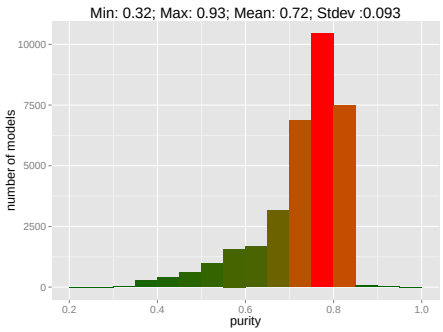

Figure C.6.31: Win.based, unreduced

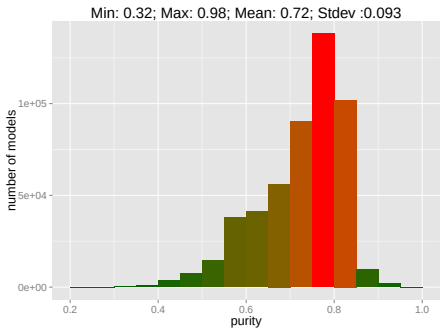

Figure C.6.32: Win.based, reduced

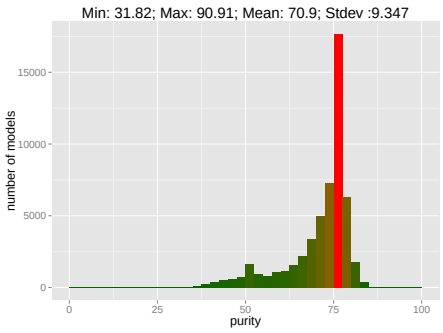

Figure C.6.33: Dep.filtered, unreduced

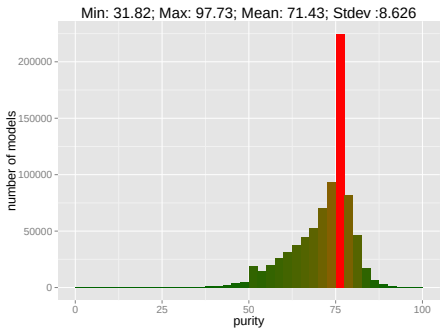

Figure C.6.34: Dep.filtered, reduced

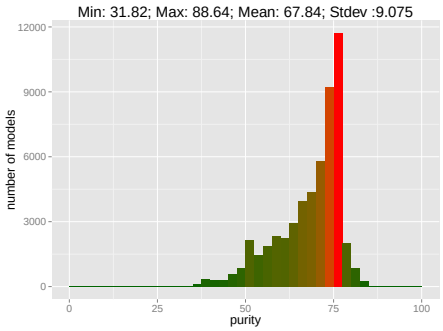

Figure C.6.35: Dep.typed, unreduced

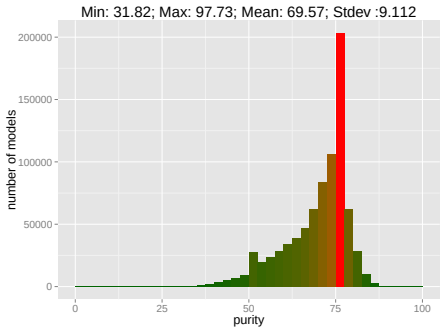

Figure C.6.36: Dep.typed, reduced

C.7 MITCHELL

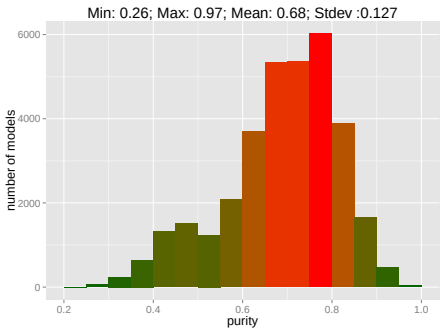

Figure C.7.37: Win.based, unreduced

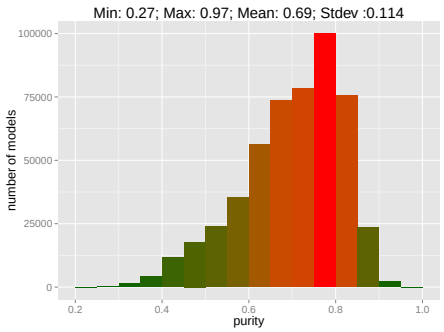

Figure C.7.38: Win.based, reduced

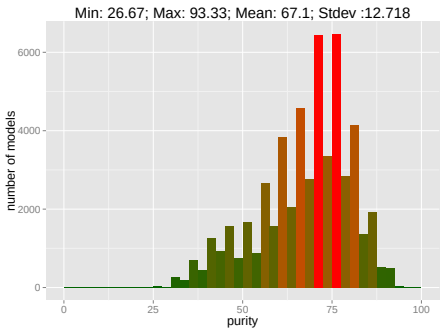

Figure C.7.39: Dep.filtered, unreduced

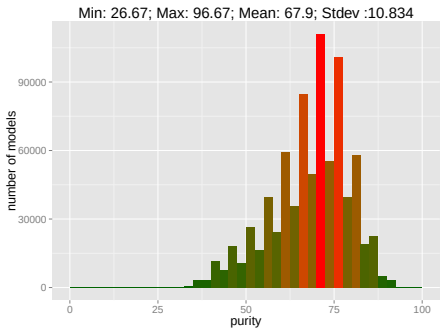

Figure C.7.40: Dep.filtered, reduced

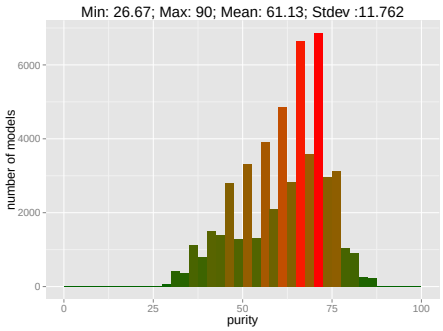

Figure C.7.41: Dep.typed, unreduced

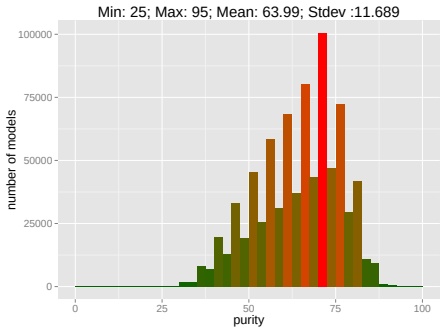

Figure C.7.42: Dep.typed, reduced

C.8 SYN

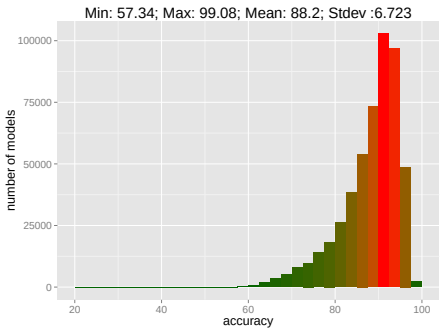

Figure C.8.43: Win.based, unreduced

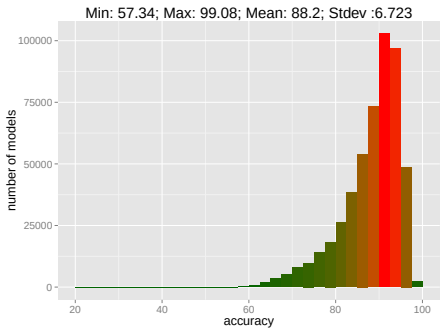

Figure C.8.44: Win.based, reduced

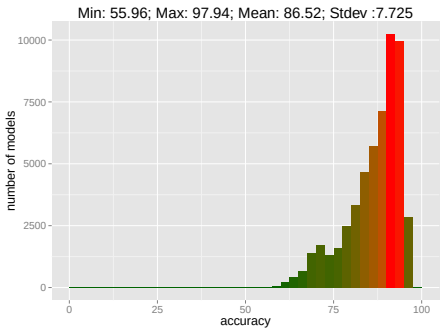

Figure C.8.45: Dep.filtered, unreduced

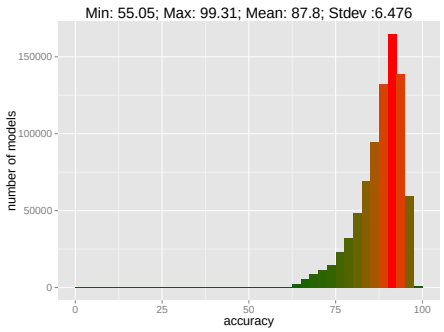

Figure C.8.46: Dep.filtered, reduced

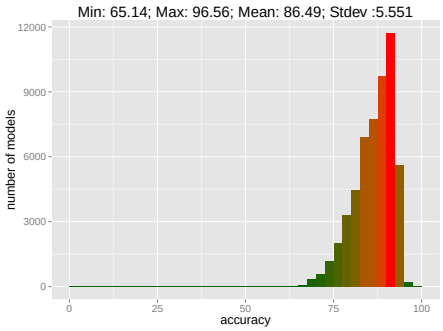

Figure C.8.47: Dep.typed, unreduced

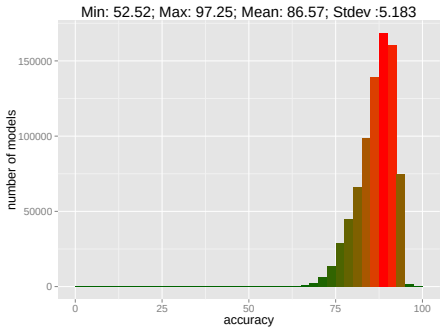

Figure C.8.48: Dep.typed, reduced

C.9 ANT

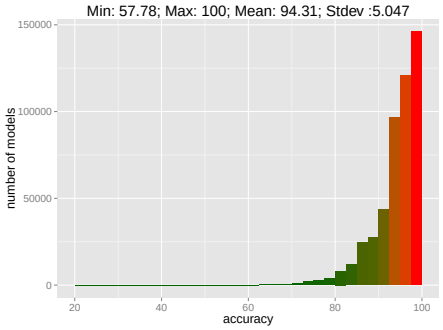

Figure C.9.49: Win.based, unreduced

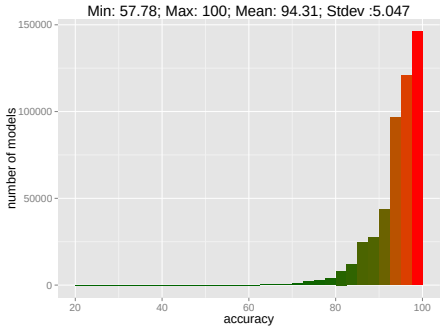

Figure C.9.50: Win.based, reduced

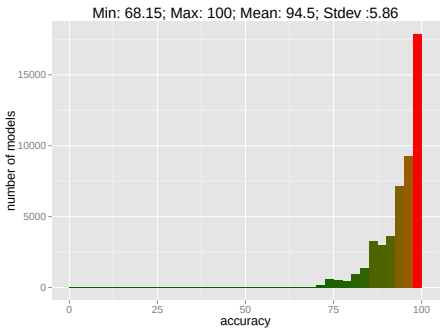

Figure C.9.51: Dep.filtered, unreduced

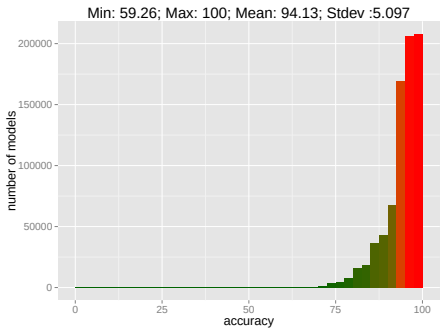

Figure C.9.52: Dep.filtered, reduced

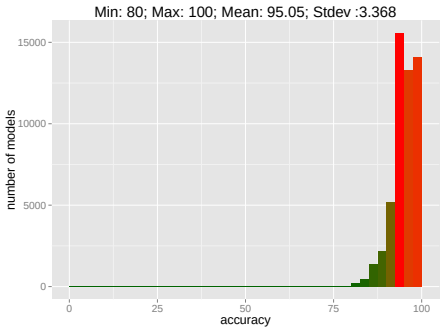

Figure C.9.53: Dep.typed, unreduced

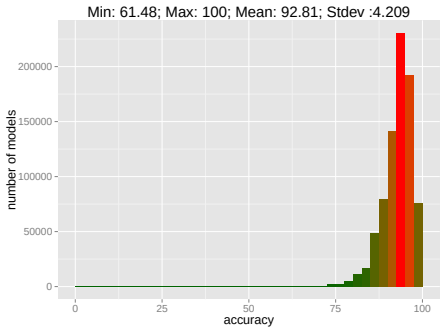

Figure C.9.54: Dep.typed, reduced

C.10 COH

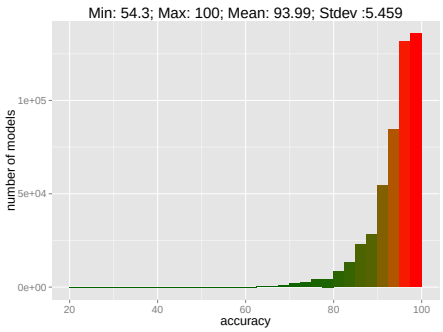

Figure C.10.55: Win.based, unreduced

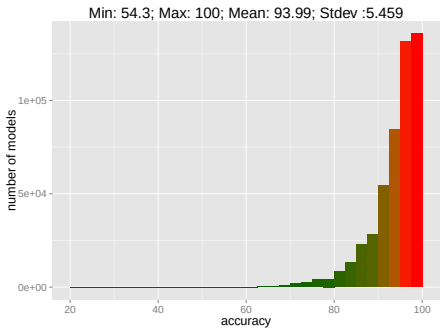

Figure C.10.56: Win.based, reduced

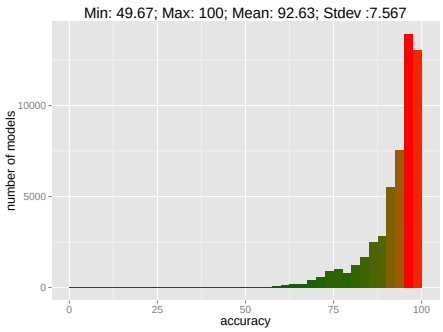

Figure C.10.57: Dep.filtered, unreduced

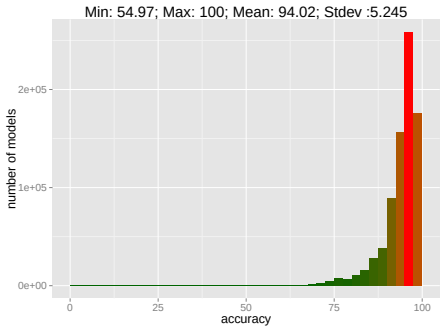

Figure C.10.58: Dep.filtered, reduced

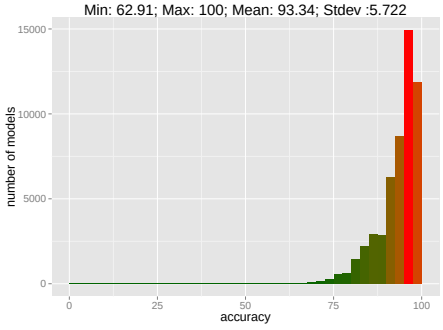

Figure C.10.59: Dep.typed, unreduced

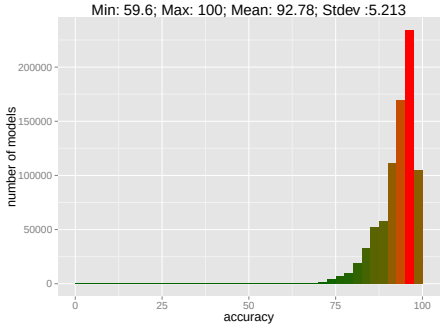

Figure C.10.60: Dep.typed, reduced

C.11 FPA

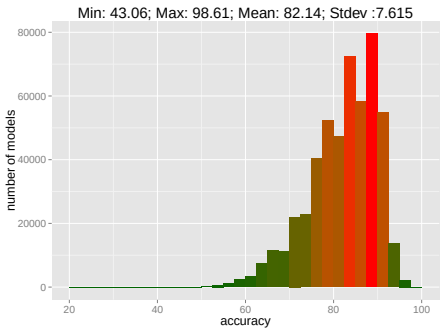

Figure C.11.61: Win.based, unreduced

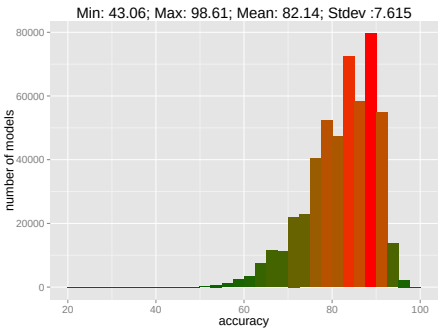

Figure C.11.62: Win.based, reduced

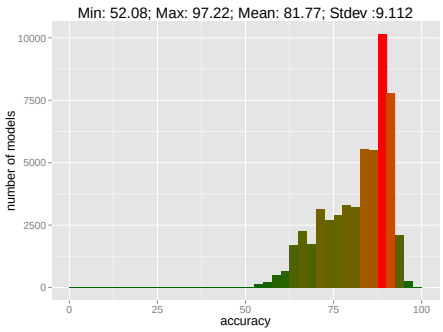

Figure C.11.63: Dep.filtered, unreduced

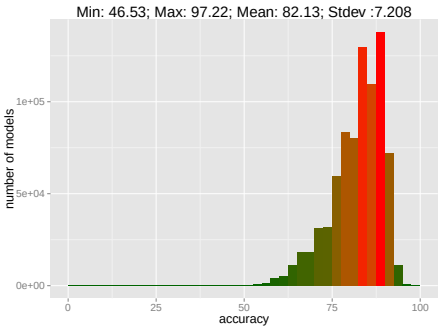

Figure C.11.64: Dep.filtered, reduced

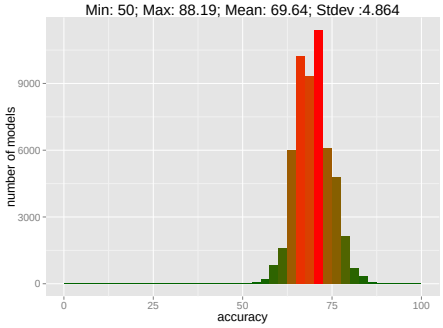

Figure C.11.65: Dep.typed, unreduced

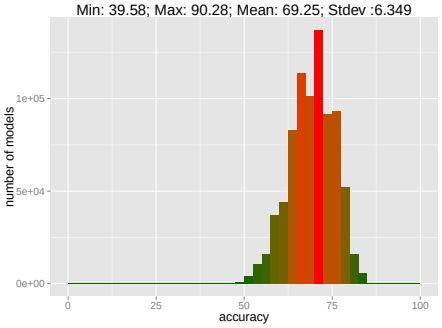

Figure C.11.66: Dep.typed, reduced

C.12 BPA

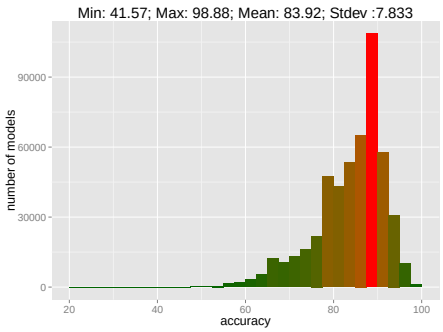

Figure C.12.67: Win.based, unreduced

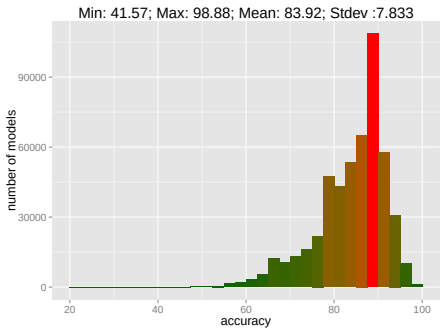

Figure C.12.68: Win.based, reduced

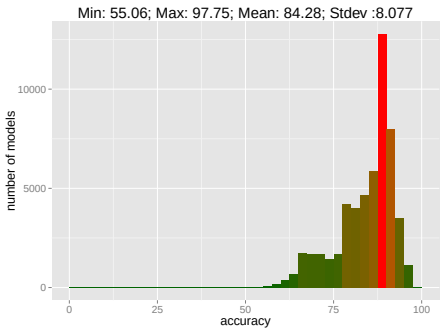

Figure C.12.69: Dep.filtered, unreduced

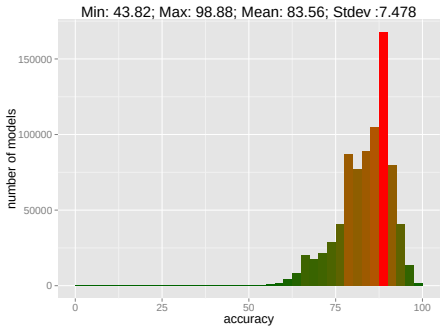

Figure C.12.70: Dep.filtered, reduced

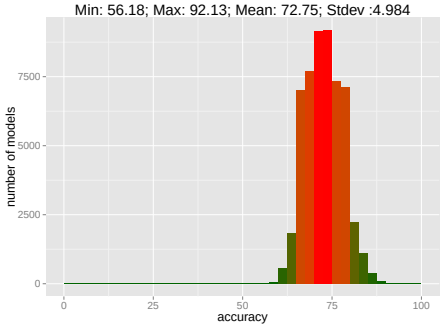

Figure C.12.71: Dep.typed, unreduced

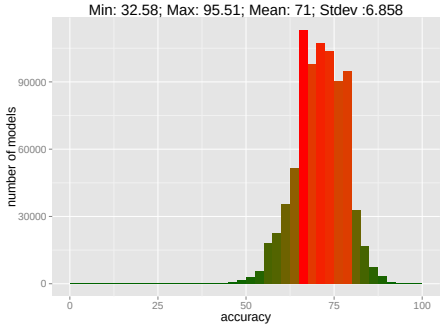

Figure C.12.72: Dep.typed, reduced

C.13 GEK

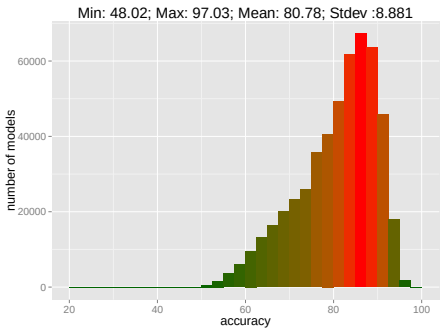

Figure C.13.73: Win.based, unreduced

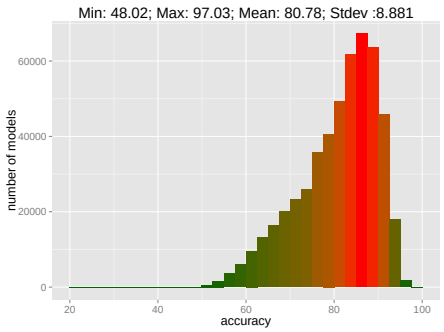

Figure C.13.74: Win.based, reduced

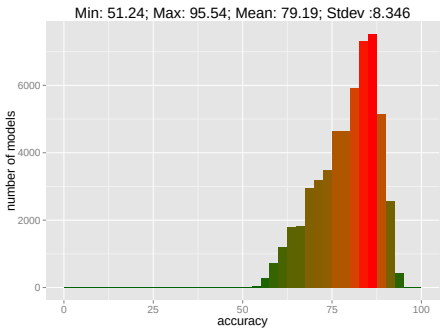

Figure C.13.75: Dep.filtered, unreduced

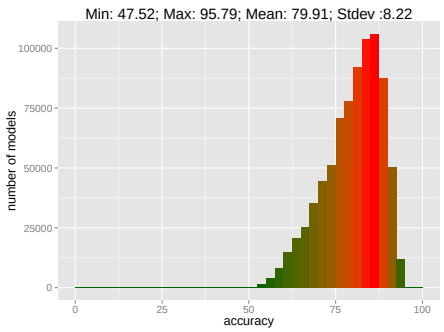

Figure C.13.76: Dep.filtered, reduced

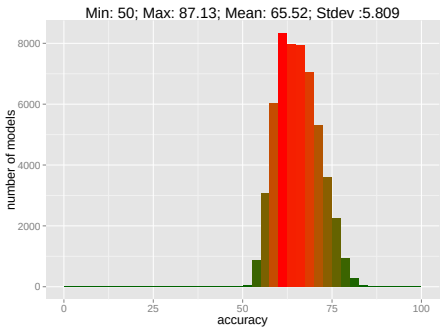

Figure C.13.77: Dep.typed, unreduced

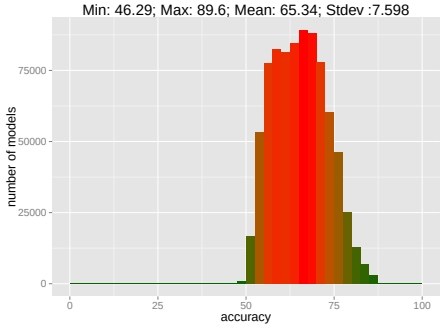

Figure C.13.78: Dep.typed, reduced

D

NaDiR: implementation details

NaDiR is designed for the multiword association task, and it contains additional features related to the particular design of the CogALex shared task (Lapesa & Evert, 2014b). NaDiR operates on lemmatized data in order to reduce sparseness. We first lemmatize the stimuli using a heuristic method, described below; then, we use the POS-annotation of the training set to train a classifier which indicates the POS of the predicted response; next, we use the predicted POS to restrict the set of response stimuli; finally, we resort to machine-learning to re-inflect the lemma thus generating a plausible word form.

Step 1: Out of context lemmatization To assign a part-of-speech tag and a lemma to every word in the dataset without relying on external tools, we adopted the following mapping strategy based on the linguistic annotation already available in UKWaC:

1. We extracted all attested wordform/part of speech/lemma combinations from UKWaC, together with their frequency;
2. Every word form in the training set was assigned to the most frequent part of speech/lemma combination attested in UKWaC.

Step 2: Prediction of the part-of-speech of the response The part-of-speech information added to every word in the dataset by the mapping procedure was used to train a classifier that, given the parts of speech of the stimuli, predicts the part of speech of the response. We trained a support-vector machine, using the `svm` function from the R package `e1071`[1], with standard settings. The part-of-speech classifier is based on a coarse part-of-speech tagset with only five tags: `N` (noun), `J` (adjective), `V` (verb), `R` (adverb), `other` (closed-class words). We considered each row of the dataset as an observation, with the part of speech of the response as predicted value, and the part of speech of the stimulus words as predictors. Every observation is represented as a bag of tags, i.e., a vector listing for each of the five tags how often it occurs among the stimuli. For example, if a set of stimuli contains 3 nouns, one verb and one adjective, the corresponding bag-of-tags vector looks as follows: {`N` = 3; `V` = 1; `J` = 1; `R` = 0; `other` = 0}. On the training set, the part-of-speech classifier achieves an accuracy of 72%.

[1] `http://cran.r-project.org/web/packages/e1071/index.html`

Step 3: Re-inflection of the predicted lemma We generate a suitable word form by inverting the heuristic lemmatization; if the full Penn tag (e.g., `NNS`: noun, common, plural; `NN`: noun, common, singular or mass, etc.) of the response is known, this step can be implemented as a deterministic lookup (since a word form is usually determined uniquely by lemma and Penn tag). We therefore trained a second SVM classifier that predicts the full Penn tag of the response based on the full tags of the stimuli. On the training set, this part-of-speech classifier reaches an accuracy of 68%.

Evaluation Table D.0.1 compares the performance of the best first-order and the best second-order model on the training and test datasets, both for lemmatized response (*Training-Lemma*, *Test-Lemma*) and generation of the correct word form (*Training-Inflected*, *Test-Inflected*).

Model	Training-Lemma	Training-Inflected	Test-Lemma	Test-Inflected
first-order	27.7% (555)	26.9% (538)	28.6% (572)	27.7% (554)
second-order	13.2% (264)	12.0% (241)	15.0% (304)	14.0% (279)

Table D.0.1: Performance (% accuracy and number of correct responses) of the best first-order and second-order model on training vs. test dataset (lemmatized response vs. response with restored inflection)

References

Almuhareb, A. (2006). *Attributes in Lexical Acquisition.* Unpublished doctoral dissertation, University of Essex.

Andrews, M., Vigliocco, G., & Vinson, D. (2009). Integrating Experiential and Distributional Data to Learn Semantic Representations. *Psychological Review*, *116*(3), 463–498.

Baayen, H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R.* Cambridge University Press.

Baroni, M., Bernardi, R., Do, N.-Q., & Shan, C.-c. (2012). Entailment Above the Word Level in Distributional Semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 23–32). Avignon, France.

Baroni, M., Bernardi, R., & Zamparelli, R. (2014). Frege in Space: A Program for Compositional Distributional Semantics. *Linguistic Issues in Language Technology*, *9*(6), 5–109.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't Count, Predict! A Systematic Comparison of Context-Counting vs. Context-Predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 238–247). Baltimore, Maryland, USA.

Baroni, M., & Lenci, A. (2008). Concepts and Properties in Word Spaces. *Italian Journal of Linguistics*, *20*(1), 55–88.

Baroni, M., & Lenci, A. (2010). Distributional Memory: A General Framework for Corpus-Based Semantics. *Computational Linguistics*, *36*(4), 1–49.

Baroni, M., & Lenci, A. (2011). How We BLESSed Distributional Semantic Evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics* (pp. 1–10). Edinburgh, UK.

Baroni, M., Murphy, B., Barbu, E., & Poesio, M. (2010). Strudel: A Corpus-Based Semantic Model Based on Properties and Types. *Cognitive Science*, *34*(2), 222–254.

Barsalou, L. W. (1999). Perceptual Symbol Systems. *Behavioral and Brain Sciences*, *22*, 513–562.

Barsalou, L. W. (2008). Grounded Cognition. *Annual Review of Psychology*, *59*, 617–645.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Boleda, G., Vecchi, E. M., Cornudella, M., & McNally, L. (2012). First-Order vs. Higher-Order Modification in Distributional Semantics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning,* (pp. 1223–1233). Jeju Island, Korea.

Bott, S., & Schulte im Walde, S. (2015). Exploiting Fine-grained Syntactic Transfer Features to Predict the Compositionality of German Particle Verbs. In *Proceedings of the 11th International Conference on Computational Semantics* (pp. 34–39). London, UK.

Brants, T., & Franz, A. (2006). *Web 1T 5-gram Version 1* (Tech. Rep.). Philadelphia, USA: Linguistic Data Consortium.

Brown, R., & Berko, J. (1960). Word Association and the Acquisition of Grammar. *Child Development*, *31*, 1–14.

Bruni, E., Boleda, G., Baroni, M., & Tran, N.-K. (2012). Distributional Semantics in Technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1* (pp. 136–145). Jeju Island, Korea.

Bruni, E., Tran, N. K., & Baroni, M. (2013). Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, *48*, 1–47.

Bullinaria, J. A., & Levy, J. P. (2007). Extracting Semantic Representations From Word Co-occurrence Statistics: A Computational Study. *Behavior Research Methods*, *39*, 510–526.

Bullinaria, J. A., & Levy, J. P. (2012). Extracting Semantic Representations From Word Co-occurrence Statistics: Stop-Lists, Stemming and SVD. *Behavior Research Methods*, *44*, 890–907.

Bullinaria, J. A., & Levy, J. P. (2013). Limiting Factors for Mapping Corpus-Based Semantic Representations to Brain Activity. *PLoS ONE*, *8*(3), 1–12.

Burgess, C., & Lund, K. (1995). *High-Dimensional Semantics From Corpora and Human Syntactic Processing Constraints.* Paper presented at the 8th Annual CUNY Sentence Processing Conference. Tucson, AZ.

Burgess, C., & Lund, K. (1998). Modeling Cerebral Asymmetries in High-Dimensional Semantic Space. In M. Beeman & C. Chiarello (Eds.), *Right Hemisphere Language Comprehension: Perspectives From Cognitive Neuroscience* (pp. 215–244). Lawrence Erlbaum Associates Publishers.

Caron, J. (2001). Experiments With LSA Scoring: Optimal Rank and Basis. In M. W. Berry (Ed.), *Computational Information Retrieval* (pp. 157–169). Philadelphia, PA: Society for Industrial and Applied Mathematics.

Chen, D., & Manning, C. (2014). A Fast and Accurate Dependency Parser Using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 740–750). Doha, Qatar.

Church, K. W., & Hanks, P. (1990). Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, *16*(1), 22–29.

Clark, H. (1970). Word Associations and Linguistic Theory. In J. Lyons (Ed.), *New Horizons in Linguistics* (pp. 271–286). Baltimore: Penguin.

Clark, S., & Curran, J. R. (2007). Wide-Coverage Efficient Statistical Parsing With CCG and Log-Linear Models. *Computational Linguistics*, *33*(4), 493–552.

Clarke, D. (2009). Context-Theoretic Semantics for Natural Language: An Overview. In *Proceedings of the EACL 2009 Workshop on GEMS: GEometical Models of Natural Language Semantics* (pp. 112–119). Athens, Greece.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (Almost) From Scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Cuba Gyllensten, A., & Sahlgren, M. (2015). Navigating the Semantic Horizon Using Relative Neighborhood Graphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2451–2460). Lisbon, Portugal.

Curran, J. R. (2003). *From Distributional to Semantic Similarity*. Unpublished doctoral dissertation, University of Edinburgh.

Curran, J. R., & Moens, M. (2002). Improvements in Automatic Thesaurus Extraction. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9* (pp. 59–66). Philadelphia, Pennsylvania, USA.

De Marneffe, M.-C., MacCartney, B., & Manning, C. D. (2006). Generating Typed Dependency Parses From Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)* (pp. 449–454). Genoa, Italy.

De Marneffe, M.-C., & Manning, C. D. (2008). *Stanford Typed Dependencies Manual (Revised for the Stanford Parser v. 3.5.1 in February 2015)* (Tech. Rep.). Stanford, USA: Stanford University.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Dinu, G., & Lapata, M. (2010). Measuring Distributional Similarity in Context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (pp. 1162–1172). Cambridge, Massachusetts, USA.

Dunning, T. (1993). Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, *19*(1), 61–74.

Edmonds, P., & Hirst, G. (2002). Near-Synonymy and Lexical Choice. *Computational Linguistics*, *28*(2), 105–144.

Erk, K., & Padó, S. (2008). A Structured Vector Space Model for Word Meaning in Context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 897–906). Honolulu, Hawaii, USA.

Erk, K., Padó, S., & Padó, U. (2010). A Flexible, Corpus-driven Model of Regular and Inverse Selectional Preferences. *Computational Linguistics*, *36*(4), 723–763.

Evert, S. (2008). Corpora and Collocations. In A. Lüdeling & M. Kytö (Eds.), *Corpus Linguistics: An International Handbook* (chap. 58). Berlin, New York: Mouton de Gruyter.

Evert, S. (2014). Distributional Semantics in R with the Wordspace Package. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations* (pp. 110–114). Dublin, Ireland.

Evert, S., & Lapesa, G. (2017). *Modelling Free Associations with Co-Occurrence Data.* Talk given at the symposium "From Computational Modelling to Behavior via Multimodal Corpus Data: Integrative Approaches to the Study of Semantic Representation and Processing". International Convention of Psychological Science. Vienna, Austria.

Ferretti, T., McRae, K., & Hatherell, A. (2001). Integrating Verbs, Situation Schemas, and Thematic Role Concepts. *Journal of Memory and Language*, *44*(4), 516–547.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, *20*(1), 116–131.

Firth, J. R. (1957). A Synopsis of Linguistic Theory 1930-1955. In *Studies in Linguistic Analysis* (pp. 1–32). The Philological Society, Oxford.

Fitzpatrick, T. (2007). Word Association Patterns: Unpacking the Assumptions. *International Journal of Applied Linguistics*, *17*(3), 319–331.

Fox, J. (2003). Effect Displays in R for Generalised Linear Models. *Journal of Statistical Software*, *8*(15), 1–27.

Frege, G. (1892). Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, *100*, 25–50.

Geffet, M., & Dagan, I. (2005). The Distributional Inclusion Hypotheses and Lexical Entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (pp. 107–114). Ann Arbor, Michigan, USA.

Ghosh, U., Jain, S., & Paul, S. (2014). A Two-Stage Approach for Computing Associative Responses to a Set of Stimulus Words. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)* (pp. 15–21). Dublin, Ireland.

Glenberg, A. M., & Robertson, D. A. (2000). Symbol Grounding and Meaning: A Comparison of High-Dimensional and Embodied Theories of Meaning. *Journal of Memory and Language*, *3*(43), 379–401.

Goldberg, Y., & Orwant, J. (2013). A Dataset of Syntactic-Ngrams Over Time From a Very Large Corpus of English Books. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity* (pp. 241–247). Atlanta, Georgia, USA.

Grefenstette, E., & Sadrzadeh, M. (2011). Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (pp. 1394–1404). Edinburgh, UK.

Grefenstette, G. (1994). *Explorations in Automatic Thesaurus Discovery.* Boston/London/Dordrecht: Kluwer Academic Publishers.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in Semantic Representation. *Psychological Review*, *114*(2), 211–244.

Halawi, G., Dror, G., Gabrilovich, E., & Koren, Y. (2012). Large-Scale Learning of Word Relatedness With Constraints. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1406–1414). Beijing, China.

Halko, N., Martinsson, P. G., & Tropp, J. A. (2011). Finding Structure With Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, *53*(2), 217–288.

Hare, M., Jones, M., Thomson, C., Kelly, S., & McRae, K. (2009). Activating Event Knowledge. *Cognition*, *111*(2), 151–167.

Harrell, F. E. J. (2015). *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis.* Springer Series in Statistics.

Harris, Z. (1954). Distributional Structure. *Word*, *10*(2–3), 146–162.

Hassan, S., & Mihalcea, R. (2011). Semantic Relatedness Using Salient Semantic Analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (pp. 884–889). San Francisco, California, USA.

Herbelot, A. (2015). Mr Darcy and Mr Toad, Gentlemen: Distributional Names and Their Kinds. In *Proceedings of the 11th International Conference on Computational Semantics* (pp. 151–161). London, UK.

Herbelot, A., & Vecchi, E. M. (2015). Building a Shared World: Mapping From Distributional to Model-Theoretic Semantic Spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 22–32). Lisbon, Portugal.

Herdağdelen, A., Erk, K., & Baroni, M. (2009). Measuring Semantic Relatedness With Vector Space Models and Random Walks. In *Proceedings of the 2009 Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-4)* (pp. 50–53). Suntec, Singapore.

Hodgson, J. (1991). Information Constraints on Pre-lexical Priming. *Language and Cognitive Processes*, *6*(3), 169–205.

Hutchison, K. A., Balota, D. A., Cortese, M. J., & Watson, J. M. (2008). Predicting Semantic Priming at the Item Level. *The Quarterly Journal of Experimental Psychology*, *61*(7), 1036–1066.

Hutchison, K. A., Balota, D. A., Neely, J. H., Cortese, M. J., Cohen-Shikora, E. R., Tse, C.-S., ... Buchanan, E. (2013). The Semantic Priming Project. *Behavior Research Methods*, *45*(4), 1099–1114.

James, W. (1890). *The Principles of Psychology.* New York: Dover.

Jones, M., & Mewhort, D. (2007). Representing Word Meaning and Order Information in a Composite Holographic Lexicon. *Psychological Review*, *114*(1), 1–37.

Joos, M. (1950). Description of Language Design. *The Journal of the Acoustical Society of America*, *22*(6), 701–708.

Jurafsky, D., & Martin, J. H. (in press). *Speech and Language Processing (3rd ed. online draft).* Prentice Hall.

Justeson, J. S., & Katz, S. M. (1992). Redefining Antonymy: The Textual Structure of a Semantic Relation. *Literary and Linguistic Computing*, *7*(3), 176–184.

Karypis, G. (2003). *CLUTO: A Clustering Toolkit (Release 2.1.1)* (Tech. Rep. No. 02-017). Minneapolis, USA: University of Minnesota, Department of Computer Science.

Katrenko, S., & Adriaans, P. (2008). Qualia Structures and Their impact on the Concrete Noun Categorization Task. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics: Bridging the Gap Between Semantic Theory and Computational Simulations* (pp. 17–24). Hamburg, Germany.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley.

Keenan, E. L., & Comrie, B. (1977). Noun Phrase Accessibility and Universal Grammar. *Linguistic Inquiry*, *8*(1), 63–99.

Kiela, D., & Clark, S. (2014). A Systematic Study of Semantic Vector Space Model Parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)* (pp. 21–30). Gothenburg, Sweden.

Kiss, G., Armstrong, C., Milroy, & Piper, J. (1973). An Associative Thesaurus of English and its Computer Analysis. In R. B. Aitken & N. Hamilton-Smith (Eds.), *The Computer and Literary Studies.* Edinburgh University Press.

Kotlerman, L., Dagan, I., Szpektor, I., & Zhitomirsky-Geffet, M. (2010). Directional Distributional Similarity for Lexical Inference. *Natural Language Engineering*, *16*(4), 359–389.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 282–289). San Francisco, California, USA.

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

Lapesa, G., & Evert, S. (2013a). Evaluating Neighbor Rank and Distance Measures as Predictors of Semantic Priming. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)* (pp. 66–74). Sofia, Bulgaria.

Lapesa, G., & Evert, S. (2013b). *Item-based Prediction of Reaction Times in Priming: an Evaluation of Distributional Semantic Models.* Poster presented at the Architecture and Mechanisms of Language Processing conference (AMLAP-2013). Marseille, France.

Lapesa, G., & Evert, S. (2013c). *Thematic Roles and Semantic Space. Insights From Distributional Semantic Models.* Paper presented at the Quantitative Investigations in Theoretical Linguistics conference (QITL-5). Leuven, Belgium.

Lapesa, G., & Evert, S. (2014a). A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection. *Transactions of the Association for Computational Linguistics*, *2*, 531–546.

Lapesa, G., & Evert, S. (2014b). NaDiR: Naive Distributional Response Generation. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)* (pp. 50–59). Dublin, Ireland.

Lapesa, G., & Evert, S. (2017). Large-Scale Evaluation of Dependency-Based DSMs: Are They Worth the Effort? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (pp. 394–400). Valencia, Spain.

Lapesa, G., Evert, S., & Schulte im Walde, S. (2014). Contrasting Syntagmatic and Paradigmatic Relations: Insights From Distributional Semantic Models. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)* (pp. 160–170). Dublin, Ireland.

Lapesa, G., Schulte im Walde, S., & Evert, S. (2014). *Judging Paradigmatic Relations: A Collection of Ratings for English.* Poster presented at the Architecture and Mechanisms of Language Processing conference (AMLAP-2014). Edinburgh, UK.

Lee, D. D., & Seung, H. S. (2000). Algorithms for Non-Negative Matrix Factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems* (pp. 556–562). Denver, Colorado, USA.

Lenci, A. (2008). Distributional Semantics in Linguistic and Cognitive Research. *Italian Journal of Linguistics*, *20*(1), 1–31.

Lenci, A., & Benotto, G. (2012). Identifying Hypernyms in Distributional Semantic Spaces. In *Proceedings of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)* (pp. 75–79). Montréal, Canada.

Levy, O., & Goldberg, Y. (2014a). Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 302–308). Baltimore, Maryland, USA.

Levy, O., & Goldberg, Y. (2014b). Neural Word Embedding as Implicit Matrix Factorization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (pp. 2177–2185). Montréal, Canada.

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving Distributional Similarity With Lessons Learned From Word Embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211–225.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 2* (pp. 768–774). Montréal, Canada.

Lin, D., & Pantel, P. (2001). Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, *7*(4), 342–360.

Lin, Y., Michel, J.-B., Aiden, E. L., Orwant, J., Brockman, W., & Petrov, S. (2012). Syntactic Annotations for the Google Books Ngram Corpus. In *Proceedings of the ACL 2012 System Demonstrations* (pp. 169–174). Jeju Island, Korea.

Lowe, W. (2001). Towards a Theory of Semantic Space. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society* (pp. 576–581). Edinburgh, UK.

Lund, K., & Burgess, C. (1996). Producing High-Dimensional Semantic Spaces From Lexical Co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*, 203–208.

MacCartney, B., & Manning, C. D. (2008). Modeling Semantic Containment and Exclusion in Natural Language Inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)* (pp. 521–528). Manchester, UK.

McCarthy, D., & Carroll, J. (2003). Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Computational Linguistics*, *29*(4), 639–654.

McDonald, S., & Brew, C. (2004). A Distributional Model of Semantic Context Effects in Lexical Processing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)* (pp. 17–24). Barcelona, Spain.

McRae, K., Cree, G. S., Seidenberg, M. S., & McNorgan, C. (2005). Semantic Feature Production Norms for a Large Set of Living and Nonliving things. *Behavior Research Methods*, *37*(4), 547–559.

McRae, K., Hare, M., Elman, J. L., & Ferretti, T. (2005). A Basis for Generating Expectancies for Verbs From Nouns. *Memory & Cognition*, *33*(7), 1174–1184.

Michel, J., Shen, Y., Aiden, A., Veres, A., Gray, M., Pickett, J., ... Aiden, E. (2011). Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, *331*(6014), 176–82.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space.* arXiv:1301.3781 [cs.CL].

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (pp. 3111–3119). Lake Tahoe, Nevada, USA.

Mikolov, T., Wen-tau, Y., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 746–751). Atlanta, Georgia, USA.

Miller, G. A., & Charles, W. G. (1991). Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, *6*(1), 1–28.

Minnen, G., Carroll, J., & Pearce, D. (2001). Applied Morphological Processing of English. *Natural Language Engineering*, *7*(3), 207–223.

Mitchell, J., & Lapata, M. (2008). Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT* (pp. 236–244). Columbus, Ohio, USA.

Mitchell, J., & Lapata, M. (2010). Composition in Distributional Models of Semantics. *Cognitive Science*, *34*, 1388–1429.

Mitchell, T., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., & Just, M. A. (2008). Predicting Human Brain Activity Associated With the Meanings of Nouns. *Science*, *320*(5880), 1191–1195.

Murdock, B. (1982). A Theory for the Storage and Retrieval of Item and Associative Information. *Psychological Review*(89), 609–626.

Murphy, B. (2002). *The Big Book of Concepts.* Cambridge: The MIT Press.

Murphy, B., Talukdar, P., & Mitchell, T. (2012). Selecting Corpus-Semantic Models for Neurolinguistic Decoding. In *Proceedings of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)* (pp. 114–123). Montréal, Canada.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (2004). The University of South Florida Free Association, Rhyme, and Word Fragment Norms. *Behavior Research Methods, Instruments, & Computers*, *36*, 402–407.

Nivre, J. (2003). An Efficient Algorithm for Projective Dependency Parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies* (pp. 149–160). Nancy, France.

Padó, S., & Lapata, M. (2007). Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, *33*(2), 161–199.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 311–318). Philadelphia, Pennsylvania, USA.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar.

Pilehvar, M. T., & Navigli, R. (2015). From Senses to Texts: An All-in-One Graph-Based Approach for Measuring Semantic Similarity. *Artificial Intelligence*, *228*, 95–128.

Plate, T. (2003). *Holographic Reduced Representation: Distributed Representation for Cognitive Structures.* CSLI Publications.

Polajnar, T., & Clark, S. (2014). Improving Distributional Semantic Vectors Through Context Selection and Normalisation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)* (pp. 230–238). Gothenburg, Sweden.

Pustejovsky, J. (1998). *The Generative Lexicon.* Cambridge, USA: MIT Press.

Rapp, R. (2002). The Computation of Word Associations: Comparing Syntagmatic and Paradigmatic Approaches. In *Proceedings of COLING 2002: The 19th International Conference on Computational Linguistics.* Taipei, Taiwan.

Rapp, R. (2013). From Stimulus to Associations and Back. In *Proceedings of the 10th Workshop on Natural Language Processing and Cognitive Science* (pp. 78–91). Marseille, France.

Rapp, R. (2014). Corpus-Based Computation of Reverse Associations. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (pp. 1380–1386). Reykjavik, Iceland.

Rapp, R., & Zock, M. (2014). The CogALex-IV Shared Task on the Lexical Access Problem. In *Proceedings of the 4th Workshop on Cognitive Aspects of the Lexicon (CogALex)* (pp. 1–14). Dublin, Ireland.

Recchia, G., Sahlgren, M., Jones, M., & Kanerva, P. (2010). Encoding Sequential Information in Semantic Space Models: Comparing Holographic Reduced Representation and Random Permutation. In *Proceedings of the 32nd Annual Cognitive Science Society* (pp. 865–870). Portland, Oregon, USA.

Rescorla, R. A., & Wagner, A. (1972). A Theory of Pavlovian Conditioning: Variations in the Effectiveness of Reinforcement and Nonreinforcement. In A. Black & W. Prokasy (Eds.), *Classical Conditioning II* (pp. 64–99).

Rohde, D. L. T., Gonnerman, L. M., & Plaut, D. C. (2006). An Improved Model of Semantic Similarity Based on Lexical Co-occurence. *Communications of the ACM*, *8*, 627–633.

Rong, X. (2014). *word2vec Parameter Learning Explained.* arXiv:1411.2738 [cs.CL].

Rothenhäusler, K., & Schütze, H. (2009). Unsupervised Classification with Dependency Based Word Spaces. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics* (pp. 17–24). Athens, Greece.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual Correlates of Synonymy. *Communications of the ACM*, *8*(10), 627–633.

Sahlgren, M. (2005). An Introduction to Random Indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005.* Copenhagen, Denmark.

Sahlgren, M. (2006). *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-Dimensional Vector Spaces.* Unpublished doctoral dissertation, University of Stockolm.

Sahlgren, M. (2008). The Distributional Hypothesis. *Italian Journal of Linguistics*, *20*(1), 33–53.

Sahlgren, M., Holst, A., & Kanerva, P. (2008). Permutations as a Means to Encode Order in Word Space. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society* (p. 1300-1305). Washington, DC, USA.

Salton, G., Wong, A., & Yang, C. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, *18*(11), 613–620.

Santus, E., Gladkova, A., Evert, S., & Lenci, A. (2016). The CogALex-V Shared Task on the Corpus-Based Identification of Semantic Relations. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)* (pp. 69–79). Osaka, Japan.

Santus, E., Lu, Q., Lenci, A., & Huang, C.-R. (2014). Unsupervised Antonym-Synonym Discrimination in Vector Space. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014* (pp. 328–332). Pisa, Italy.

Scheible, S., Schulte im Walde, S., & Springorum, S. (2013). Uncovering Distributional Differences Between Synonyms and Antonyms in a Word Space Model. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (pp. 489–497). Nagoya, Japan.

Schmid, H. (1995). Improvements in Part-of-Speech Tagging With an Application to German. In *Proceedings of the ACL SIGDAT-Workshop* (pp. 47–50). Dublin, Ireland.

Schuster, S., & Manning, C. D. (2016). Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 2371–2378). Portorož, Slovenia.

Schütze, H. (1992). Dimensions of Meaning. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing* (pp. 787–796). Minneapolis, Minnesota, USA.

Schütze, H. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, *27*(1), 97–123.

Speer, R., Chin, J., & Havasi, C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4444–4451). San Francisco, CA, USA.

Sridharan, S., & Murphy, B. (2012). Modeling Word Meaning: Distributional Semantics and the Corpus Quality-Quantity Trade-Off. In *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon* (pp. 53–68). Mumbai, India.

Stone, B. P., Dennis, S. J., & Kwantes, P. J. (2008). A Systematic Comparison of Semantic Models on Human Similarity Rating Data. The Effectiveness of Subspacing. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 1813–1818). Austin, Texas, USA.

Terra, E., & Clarke, C. (2003). Frequency Estimates for Statistical Word Similarity Measures. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 244–251). Edmonton, Canada.

Thater, S., Fürstenau, H., & Pinkal, M. (2011). Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of the 5th International Joint Conference on Natural Language Processing* (pp. 1134–1143). Chiang Mai, Thailand.

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)* (pp. 491–502). Freiburg, Germany.

Turney, P. D. (2006). Similarity of Semantic Relations. *Computational Linguistics*, *32*(3), 379–416.

Turney, P. D. (2008). A Uniform Approach to Analogies, Synonyms, Antonyms, and Associations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)* (pp. 905–912). Manchester, UK.

Turney, P. D., & Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, *37*(1), 141–188.

Tversky, A. (1977). Features of Similarity. *Psychological Review*, *84*(4), 327–352.

Utt, J., & Padó, S. (2014). Crosslingual and Multilingual Construction of Syntax-Based Vector Space Models. *Transactions of the Association for Computational Linguistics*, *2*, 245–258.

Van De Cruys, T. (2010). A Non-Negative Tensor Factorization Model for Selectional Preference Induction. *Natural Language Engineering*, *16*(4), 417–437.

Van Overschelde, J., Rawson, K., & Dunlosky, J. (2004). Category Norms: An Updated and Expanded Version of the Battig and Montague (1969) Norms. *Journal of Memory and Language*, *50*(3), 289–335.

Vecchi, E. M., Baroni, M., & Zamparelli, R. (2011). (Linear) Maps of the Impossible: Capturing Semantic Anomalies in Distributional Space. In *Proceedings of the Workshop on Distributional Semantics and Compositionality* (pp. 1–9). Portland, Oregon, USA.

Vinson, D., Andrew, M., & Vigliocco, G. (2013). Giving Words Meaning: Why Better Models of Semantics are Needed in Language Production Research. In M. Goldrick, V. S. Ferreira, & M. Miozzo (Eds.), *Oxford Handbook of Language Production* (pp. 134–151). Oxford University Press.

Weeds, J., & Weir, D. (2003). A General Framework for Distributional Similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (pp. 81–88). Sapporo, Japan.

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising Measures of Lexical Distributional Similarity. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)* (pp. 1015–1021). Geneva, Switzerland.

Wettler, M., Rapp, R., & Sedlmeier, P. (2005). Free Word Associations Correspond to Contiguities Between Words in Texts. *Journal of Quantitative Linguistics*, *12*(2–3), 111–122.

Widdows, D. (2004). *Geometry and Meaning*. CSLI publications.

Zeller, B., Padó, S., & Šnajder, J. (2014). Towards Semantic Validation of a Derivational Lexicon. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 1728–1739). Dublin, Ireland.

Zhang, Y., & Clark, S. (2008). A Tale of Two Parsers: Investigating and Combining Graph-Based and Transition-Based Dependency Parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (pp. 562–571). Honolulu, Hawaii, USA.