Computational approaches for German particle verbs: compositionality, sense discrimination and non-literal language

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Philosophie (Dr. phil.) genehmigte Abhandlung.

Vorgelegt von

Maximilian Köper

aus Esslingen am Neckar

Hauptberichter	PD Dr. Sabine Schulte im Walde
Mitberichter	Prof. Dr. Sebastian Padó
Mitberichter	Dr. Peter D. Turney

Tag der mündlichen Prüfung: 14.09.2018

Institut für Maschinelle Sprachverarbeitung
der Universität Stuttgart

2018

Abstract

Anfangen *(to start)* is a German particle verb. Consisting of two parts, a base verb ("fangen") and particle ("an"), with potentially many or no intervening words in a sentence, particle verbs are highly frequent constructions with special properties.

It has been shown that this type of verb represents a serious problem for language technology, due to particle verbs' ambiguity, ability to occur separate and seemingly unpredictable behaviour in terms of meaning. This dissertation addresses the meaning of German particle verbs via large-scale computational approaches. The three central parts of the thesis are concerned with computational models for the following components: i) compositionality, ii) senses and iii) non-literal language. In the first part of this thesis, we shed light on the phenomena by providing information on the properties of particle verbs, as well as the related and prior literature. In addition, we present the first corpus-driven statistical analysis.

We use two different approaches for addressing the modelling of compositionality. For both approaches, we rely on large amounts of textual data with an algebraic model for representation to approximate meaning. We put forward the existing methodology and show that the prediction of compositionality can be improved by considering visual information.

We model the particle verb senses based only on huge amounts of texts, without access to other resources. Furthermore, we compare and introduce the methods to find and represent different verb senses. Our findings indicate the usefulness of such sense-specific models.

We successfully present the first model for detecting the non-literal language of particle verbs in a running text. Our approach reaches high performance by combining the established techniques from metaphor detection with particle verb-specific information.

In the last part of the thesis, we approach the regularities and the meaning shift patterns. Here, we introduce a novel data collection approach for accessing the meaning components, as well as a computational model of particle verb analogy. The experiments reveal typical patterns in domain changes. Our data collection indicates that coherent verbs with the same meaning shift represent rather scarce phenomena.

In summary, we provide novel computational models to previously unaddressed problems, and we report incremental improvements in the existing approaches. Across the models, we observe that semantically similar or synonymous base verbs behave similarly when combined with a particle. In addition, our models demonstrate the difficulty of particle verbs. Finally, our experiments suggest the usefulness of external normative emotion and affect ratings.

Deutsche Zusammenfassung

Partikelverben sind außergewöhnliche und gleichzeitig häufig auftretende Konstrukte. Sie bestehen aus zwei Teilen, so enthält das Partikelverb *anfangen* das Basisverb (“fangen”) sowie die Partikel (“an”). Darüber hinaus können Partikelverben zusammengeschrieben oder getrennt in einem Satz erscheinen.

Im Bereich der Computerlinguistik stellen Partikelverben aufgrund ihrer idiosynkratischen Eigenschaften eine große Herausforderung dar. So besitzen sie in der Regel eine Vielzahl an Bedeutungen. Darüber hinaus weisen sie unterschiedliche Grade von Kompositionalität im Hinblick auf ihre Konstituenten auf.

In der vorliegenden Dissertation wird die Bedeutung deutscher Partikelverben umfassend anhand komputationeller Modelle behandelt. Die drei zentralen Themen der Arbeit sind i) Kompositionalität ii) Bedeutungsunterscheidung sowie iii) Nicht-wörtliche Sprache. Der erste Teil der Arbeit präsentiert zunächst Eigenschaften von Partikelverben sowie relevante Literatur. Des Weiteren präsentieren wir eine erste statistische Korpusauswertung zum Phänomen der Partikelverben.

Für die Modellierung von Kompositionalität verfolgen wir zwei Ansätze. Beide Ansätze verwenden große Mengen geschriebener Sprache sowie ein mathematisches Vektorraummodell, um Wortbedeutung zu repräsentieren. Unsere Experimente zeigen, dass die Vorhersage von Kompositionalitätsbewertungen durch zusätzliche visuelle Information verbessert werden kann.

Des Weiteren vergleichen wir bestehende und präsentieren neue Methoden, die in der Lage sind, verschiedene Wortbedeutungen zu finden und diese darzustellen. Unsere Ergebnisse unterstreichen den Nutzen von Modellen, die unterschiedliche Lesarten separat darstellen.

Darüber hinaus liefert diese Arbeit das erste Modell zur automatischen Erkennung von nicht-wörtlicher Verwendung deutscher Partikelverben. Durch die Kombination von bewährten Techniken aus dem Bereich der automatischen Metapher Erkennung sowie neuen, Partikelverb-spezifischen Informationen, erzielt unser Modell hohe Genauigkeit.

Der letzte Teil dieser Arbeit behandelt Muster und reguläre Bedeutungsveränderungen. Wir verwenden hierbei eine neue Domain-Datensammlung sowie ein komputationelles Analogiemodell. Unsere Experimente verdeutlichen häufige Muster für Domainveränderungen und zeigen, dass reguläre Bedeutungsveränderung zwischen kohärenten Verben ein eher seltenes Phänomen darstellen.

Zusammenfassend präsentiert diese Arbeit komputationelle Modelle für zuvor nicht berücksichtigte Fragestellungen und verbessert bisherige Ansätze zur modellierung deutscher Partikelverben. Wir beobachten über verschiedene Modelle hinweg, dass semantisch ähnliche Basisverben ein ähnliches Verhalten zeigen, wenn sie mit einer Partikel kombiniert werden. Ebenso demonstrieren unsere Experimente die besondere Schwierigkeit von Partikelverben und unterstreichen den Nutzen von externen Emotions oder Affekt Beurteilungen.

Acknowledgments

I would like to thank my adviser, Sabine Schulte im Walde. Without her, I would have had nearly zero publications and this thesis would not have been written. Sabine has given me the support and guidance, as well as the space and freedom, to define my research and develop as a researcher in the best possible way.

I would also like to thank Sebastian Padó and Peter Turney for being part of the doctoral committee. Thank you for investing time and providing valuable feedback. I feel proud and honoured that you have agreed to be on my committee.

Over the last several years I have greatly benefitted from teachers, and later colleagues, at IMS. I consider myself lucky to work in such a productive and friendly research environment. I want to express special thanks to my colleagues, Anders Bjoerkelund, Michael Walsh, Chrisitan Scheible, Daniela Naumann, Dominik Schlechtweg, Kyle Richardson, Ngoc Thang Vu, Roman Klinger and Stephen Roller. Beyond the IMS, I had the chance to learn and join forces in fruitful collaborations. I thank Steffen Koch for the expertise on visualization. Thanks to Eleri Aedmaa for her knowledge and expertise on Estonian particle verbs.

I would also like to thank the people behind the scenes. Thanks to the various annotators, involved in my experiments. I am grateful to Edar Hoch and the system administration crew for the technical assistance, as well as Sybille Laderer and Sabine Mohr for solving administrative issues. I gratefully acknowledge the DFG for providing the funding of our project D12 within the SFB-732.

It has been a great pleasure to have shared my office with different people over the last years. Thanks to Sai Abishek Bhaskar, Nana Khvtisavrishvili, Evi Kiagia and Sylvia Springorum.

Special thanks go to the best next-door colleagues. I will definitely miss the discussions during the countless coffee breaks that I enjoyed with Stefan Bott, Jeremy Barnes and Kim-Anh Nguyen.

I had the endless support and encouragement of my parents during my studies. I am also grateful to my siblings and friends outside of computational linguistics for providing me with the necessary distractions from my research.

My greatest appreciation goes to my wife, Patricia, and our two lovely kids, Emma and Niklas. Thanks for the patience and support.

Contents

List of Abbreviations

AI	Artificial intelligence
ANET	AlexNet
BoVW	Bag of Visual Words
BV	Base Verb(s)
CNN	Convolutional Neural Network
DSMs	distributional semantic models(s)
FCM	Fuzzy C-Means
GNET	GoogLeNet
HDP	Hierarchical Dirichlet Process
LDA	Latent Dirichlet Allocation
LMI	Local Mutual Information
LS	Local Scaling
NLP	Natural Language Processing
NI	Non-Iterative Contextual Measure
MNB	Multinomial Naive Bayes
MWE	Multiword Expression(s)
PMI	Pointwise Mutual Information
PPMI	Positive Pointwise Mutual Information
PV	Particle Verb(s)
SGNS	Skip Gram with Negative Sampling
SVD	Singular Value Decomposition
SVMs	Support Vector Machines
VSM	Vector Space Model
WSD	Word Sense Disambiguation

List of Tables

List of Figures

1

Introduction

Languages are made up of words, which combine to form sentences and interact to form structures that convey meaning. However, the notions of word and word meaning are surprisingly complex and difficult to pin down. Particle verbs (PVs) are complex constructions combining the properties of words and syntactic phrases. Briefly, these verbs consist of a base verb (BV) and, roughly speaking, some pre-verb or particle. As an example, the German particle "auf" can be combined with the verb "sammeln" *(to gather)* to build the PV "auf+sammeln" *(to gather up)*.

Due to their high productivity, PVs are ubiquitous in the German language, and they present a fundamental tool for word formation. Beyond their ability to occur syntactically separated, PVs possess a range of surprising and unpredictable properties. For instance, these verbs can be compositional. Here, the meaning of the whole PV can be determined by the meanings of its constituents, as demonstrated by "kleben" *(to stick)* combined with "an" resulting in the PV "ankleben" *(to stick on)*. This is in contrast to non-transparent or opaque PV constructions with meanings that cannot be inferred easily from their individual parts, such as "fangen" *(to catch)* combined with "an", resulting in "anfangen", meaning *to start*.

That aside, PVs are likely to carry various meanings, since both their constituents (i.e., particles and base verbs) may be highly ambiguous. Consequently, it is likely that a PV, as a combination of these ambiguous parts, will also be ambiguous. Furthermore, often, the particles trigger meaning shifts when they combine with base verbs; therefore, the resulting PVs are frequent cases of non-literal meaning. For example, there are at least two senses of "aufsprudeln", of which one is literal, meaning "*bubble up*", and one results from a meaning shift, "*become angry*". It is hypothesized that such meaning shifts are frequently regular, and they can be applied across a set of coherent base verbs, for example, the two verbs "aufkochen" and "aufbrausen" share the same shifted sense as "aufsprudeln".

Humans (native speakers) are able to produce and understand such verbs effortlessly. In addition, a computational model of language incapable of performing such tasks. Hence, numerous researchers have pointed out the difficulty and the importance of such constructions. Consequently, they represent a serious and challenging problem for natural language processing (NLP).

PVs are problematic in terms of at least two key issues for NLP. The first is disambiguation, that is, finding and determining the correct meaning of a verb. The second problem is that PVs cross word boundaries and belong to the challenging class of multiword expressions (MWEs). Regarding concrete applications, PVs require special treatment across a broad range of fields, such as automatic translation, parsing, information and terminology extraction, and natural language understanding.

In short, in the words of Sag et al. (2002), PVs are *"A Pain in the Neck for NLP"*.

At the same time, the characteristics that make this class of verbs challenging and seemingly unpredictable are fascinating and make them an interesting subject.

1.1 Motivation and Research Questions

NLP, as a subfield of artificial intelligence (AI), bridges the gap between computers and human language. Recent advances in NLP have led to sophisticated applications that have entered our everyday life. Users encounter such devices with high expectations about their ability to communicate. However, such devices only imitate intelligence and mimic language; they have little understanding of the actual meaning of human language. Even an assistant equipped with the knowledge of morphology, syntax, and the semantics of individual words would still fail when confronted with highly ambiguous MWEs. Being such an obstacle for language technology makes PVs an especially interesting phenomenon to research.

The purpose of the application aside, little is known about the underlying phenomena or mechanisms that trigger novel senses or meaning shifts when a particle is combined with a BV. Theoretical work on PVs has focused predominantly on the question of whether PVs should be treated as instances of words (morphological objects) or syntactic combinations. More semantically motivated approaches typically focus on small target sets, usually restricted to a single particle, a single type of particle contribution (e.g., spatial) or only a handful of PV constructions. In contrast, statistical approaches and computational models represent promising tools that are capable of addressing PV semantics automatically and on a large scale. Such models can give new insights into the phenomenon.

This thesis aims to conduct a large-scale study dealing with multiple particles and PVs in relation to the following three main challenges: the compositionality, sense discrimination and non-literal usage of German PVs. In addition, we[1] exploit the existence of regular meaning shifts from one domain to another when a BV is combined with a particle.

The main focus of this thesis is computational modeling. Using the increasing availability of massive amounts of textual data, we rely on huge web corpora as underlying information for our computational approaches. The main tools for the computational models are statistical approaches. We exploit collocations, machine learning and a representational framework, namely distributional semantics. Here, we utilize the distributional properties of words to approximate their meaning and estimate their similarities.

Concerning compositionality, we exploit the extent to which we can build a model that determines the degree of compositionality between a PV and its BV. Furthermore, we are interested in identifying the salient features for this task, and we address the question of whether a purely distributional model can be enhanced by taking additional perceptual information into account. In addition, we model the contribution of the particle in a PV construction by relying on the methods from compositional distributional semantics.

With respect to sense discrimination, the thesis investigates the use of the methods to learn a distinct representation for each verb sense. Here, we focus on unsupervised methods that require no external resources. We address this problem on a type-based level. Furthermore, we investigate the usefulness of such models in the context of PV-specific tasks. Equally importantly, we are interested in how such methods compare against traditional representations, where multiple senses are stored in a single representation.

For non-literal language, we are interested in a model that detects literal versus non-literal usage of PVs automatically. This phenomenon is addressed on a token-based level. Here, we are interested in the indicators for non-literal language and the question of whether we can facilitate PV-specific information and the standard features from metaphor detection for this task. Furthermore, we are interested in regularities with respect to particle or semantically similar BVs.

Finally, we investigate the existence of regular meaning shifts. Here, we identify typical patterns from BV to non-literal PV usage. Second, by applying a computa-

[1] Although this thesis was written by a single author, I do not favor the first person "I" form in scientific texts. Therefore, I use "we" rather than "I".

tional model of analogy, we distinguish various types of meaning shifts.

1.2 Thesis Structure

In brief, the structure of this thesis follows the order of topics in the title of the presented thesis. Hence, after providing information on the theory and the methods, we first address the modeling of compositionality, then, sense discrimination, and further non-literal language. Following this introduction chapter, two further chapters are warranted before we arrive at computational approaches, as outlined below.

Chapter 2 provides background information on the various research directions. Here, Section 2.1 illustrates the phenomena of PVs in more detail. In addition, an overview of the related theoretical and computational work on the subject is given. Next, Section 2.2 provides the necessary background information and covers the broad topic of ambiguity and sense discrimination. Similarly, Section 2.3 provides the theoretical background for the topic of non-literal language.

Chapter 3 provides background information on the methods and resources used. In this chapter, first, the concept of distributional semantics is introduced (Section 3.1). Next, there is a brief description of the various machine learning methods and evaluation metrics used in this thesis (Section 3.2). Since multiple experiments make heavy usage of abstractness and concreteness, as well as affective norms, we describe these concepts in Section 3.3. Section 3.4 continues with the methods of automatically extending such norms to larger dictionaries. Concerning resources, Section 3.5 describes the commonly used underlying corpora. Subsection 3.5.2 uses these corpora to present a corpus study on the phenomenon of PVs.

Chapter 4 presents computational approaches for modeling compositionality. This chapter consists of three different experiments: Section 4.1 makes use of previously defined concepts and resources. Here, a distributional model is used to predict compositionality. Section 4.2 extends this model with visual information. Finally, the last experiment in this chapter models PV compositionality as a vector operation (Section 4.3).

Chapter 5 addresses the modeling of PV senses. This is done by first exploring a token-based classification between the literal and non-literal usage of PVs in Section 5.1. Beyond this binary distinction, a type-based multi-sense modeling is performed in Section 5.2.

Chapter 6 is concerned with regular behavior and patterns with respect to PV meaning shifts. Section 6.1 presents a sentence collection with annotated domains to exploit common domain changes from literal BV usage to shifted PV usage. Section 6.2 presents an analogy dataset and a computational model of PV analogy for detecting the different kinds of meaning shifts.

Finally, **Chapter 7** summarizes the main findings and results. Section 7.2 discusses potential directions for future research.

1.3 Contributions

The main objective of this thesis is to model German PVs, especially the phenomena of compositionality, senses and non-literal language. While we contribute considerably these topics, research often leads to further findings and contributions in other directions. Therefore, we provide a list of the main contributions of the thesis below, divided according to the nature or topic of the respective contribution.

Modeling Compositionality: We systematically study various distributional models to predict compositionality for German PVs. Our findings show the importance of reconstruction solutions that account for syntactically separated PVs. Further, we show that adding visual information to a textual model can improve the prediction of compositionality. Our findings suggest that such multi-model approaches should rely on external imageability norms.

We model the contribution of the particle in a compositional distributional semantic setup. Here, we observe that PV-motivated training-space restrictions enhance our models. Moreover, particle-specific explorations show differences across particle types and demonstrate the difficulty of such constructions in contrast to other derivations.

Modeling Senses: We address PV senses by applying a type-based approach. Here, we investigate several variants of state-of-the-art methods for obtaining multi-sense representations. We investigate the quality of these representations by using them for various PV-specific semantic tasks, such as semantic verb classification, the prediction of compositionality, and the detection of non-literal language. Our findings confirm the need to distinguish between PV senses in a distributional semantic model.

Non-Literal Language of PVs: We present the first computational model that addresses the literal versus non-literal language usage of German PVs. Here, we exploit a variety of traditional features, as well as a novel PV-specific distributional fit feature. The classifier significantly outperforms a majority baseline by reaching a maximum accuracy of 86.8%. We demonstrate that PVs with semantically similar particles and semantically similar BVs can predict each others' literal vs. non-literal language usage.

Regular Meaning Shifts: We conduct a type-level and token-level experiment to study the meaning shifts of German PVs. By aligning BV and PV sentences to the source and respective target domains, we can study typical changes from the Source→Target domains. Using statistical counts and association strength measures, we detect patterns that commonly apply when a BV is combined with a particle. These patterns reflect theories regarding metaphor detection.

Relying on a novel analogy dataset of BV-PV pairs, we observe that regular meaning shifts, where both BVs belong to a semantically coherent set of verbs, are rather infrequent. Furthermore, we present a computational model of analogy to distinguish the different kinds of meaning shifts. Here, we show that affective or emotional information represents the most salient indicator.

PV Statistics: We present the first large-scale corpus-based analysis of PVs. We provide new insights by zooming into PV-specific properties, such as particle frequency, separability, separability-distance and sense information.

Methodology: In this thesis, we present novel methods and experimental setups that are potentially applicable to a variety of other research questions. For the methods, one contribution comprises our novel techniques for multi-modal distributional semantics. Here, we successfully incorporate imageability norms and exploit novel clustering techniques to enhance the resulting verb representation.

Next, we successfully apply the local scaling method, which was originally used in music retrieval, to mitigate the hubness problem in a nearest-neighbor search for words. Furthermore, we introduce two novel methods to perform multi-sense representation learning by utilizing non-parametric clustering techniques. These techniques perform both sense induction and sense disambiguation. Here, we also demonstrate that multi-sense representation in combination with hard-clustering can be seen as an alternative and a promising method to perform soft-clustering. In addition, in a variety of experiments, we extend the standard distributional information with affect and emotion ratings to access information beyond pure textual models.

Concerning the experimental setup, we conduct experiments with uncommon and new techniques to gain better and novel insights. For the evaluation of multiple models with many parameters, we present the results in terms of score distributions that support the robustness of our findings. Another contribution, is our (non-literal) classification setup. Here, we zoom into particle or verb-specific behavior by restricting training and evaluation data according to the properties of interest. While previous models on (verb) classification focused strongly on hard assignments, we overcome this limitation and propose experimental setups to perform and evaluate the quality of the more challenging, but at the same time more realistic, soft assignments (soft clustering).

PV Datasets: We present a variety of novel datasets that can be used for future work on German PVs. These datasets contain a new collection of German PV derivations, comprising 1410 BV→PV patterns across seven particles. Furthermore, we create a large collection of 6436 German sentences annotated by three annotators for literal versus non-literal usage across 10 particles and 159 PVs.

We present a collection and a novel strategy for obtaining source and target-domain characterizations. The resulting dataset contains 7420 sentences of 138 German BVs and their 323 existing PVs annotated for various source (BV) and target (PV) domains. In addition, the sentences contain non-literalness scores and directionality information.

We also collect a novel analogy dataset. This resource contains 794 analogies annotated according to four different kinds of meaning shifts.

Affective Norms: In addressing various research questions, we successfully exploit affective norm information in our computational approaches. To incorporate this information, we conduct experiments with methods learning and extending such norms. Here, our contribution is two-fold: First, we introduce novel techniques to automatically extend such ratings norms to phrases and senses, as well as across languages. This methodology can be applied to any language. Second, by applying these techniques, we create a considerable amount of affective norms for German and Estonian, where these norms did not exist in such numbers before.

1.4 Publications

Parts of the research in this thesis have been published in the list below. It is clear from the references that the research was usually done in collaboration. This is another reason why I prefer to use "we" rather than "I" in this thesis. The thesis will focus on the work where I was first author (see the following list), except for the paper described in Schulte im Walde et al. (2018); for the latter case, my contribution is stated below.

- Köper, M. and Schulte im Walde, S. (2016). Automatically Generated Affective Norms of Abstractness, Arousal, Imageability and Valence for 350 000 German Lemmas. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 2595–2598, Portoroz, Slovenia

- Köper, M. and Schulte im Walde, S. (2016). Distinguishing Literal and Non-Literal Usage of German Particle Verbs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 353–362, San Diego, California, USA

- Köper, M., Schulte im Walde, S., Kisselew, M., and Padó, S. (2016). Improving Zero-Shot-Learning for German Particle Verbs by using Training-Space Restrictions and Local Scaling. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 91–96, Berlin, Germany

- Köper, M. and Schulte im Walde, S. (2017a). Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 535–542, Valencia, Spain

- Köper, M. and Schulte im Walde, S. (2017b). Complex Verbs are Different: Exploring the Visual Modality in Multi-Modal Models to Predict Compositionality. In *Proceedings of the 13th Workshop on Multiword Expressions*, pages 200–206, Valencia, Spain

- Köper, M. and Schulte im Walde, S. (2018). Analogies in Complex Verb Meaning Shifts: The Effect of Affect in Semantic Similarity Models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 150–156, New Orleans, Louisiana , USA

In addition, the thesis contains published experiments, namely the Source→Target mapping, of the publication below. Sabine Schulte im Walde designed, supervised and performed the qualitative analysis of the data collection. My contribution was the computational modeling, namely, performing the classification experiments, the statistical analysis, and the data visualization of the collection.

- Schulte im Walde, S., Köper, M., and Springorum, S. (2018). Assessing Meaning Components in German Complex Verbs: A Collection of Source-Target Domains and Directionality. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 22–32, New Orleans, LA, USA

The following work is relevant to the subject of this thesis but will not be fully explicated. Concerning the work in Wittmann et al. (2017), my role involved the creation of word representations and I helped with the setup of the soft-clustering method. Regarding the publication in Aedmaa et al. (2018), I was responsible for the creation of Estonian abstractness norms and providing advice for the machine learning usage in the experimental part of the research.

- Wittmann, M., Köper, M., and Schulte im Walde, S. (2017). Exploring Soft-Clustering for German (Particle) Verbs across Frequency Ranges. In *Proceedings of the 12th International Conference on Computational Semantics*, Montpellier, France

- Aedmaa, E., Köper, M., and Schulte im Walde, S. (2018). Combining Abstractness and Language-specific Theoretical Indicators for Detecting Non-Literal Usage of Estonian Particle Verbs. In *Proceedings of the NAACL Student Research Workshop*, pages 117–218

2 Phenomena / Theoretical Background

2.1 Particle Verbs

> “*The Germans have another kind of parenthesis, which they make by splitting a verb in two and putting half of it at the beginning of an exciting chapter and the other half at the end of it. Can any one conceive of anything more confusing than that? These things are called separable verbs. The German grammar is blistered all over with separable verbs; and the wider the two portions of one of them are spread apart, the better the author of the crime is pleased with his performance.*
>
> Mark Twain, *The Awful German Language. Appendix D*, 1880”

The literature contains many names for PVs: phrasal verbs, verb-particle combinations, separable verbs, and complex verbs. Particle verbs, such as “aufgeben” *(to give up)*, “aufhören” *(to cease)*, or “anlachen” *(to laugh at)* are constructions consisting of a base verb and a particle (sometimes preverb). Particle verbs occur in all Germanic languages (Dehé, 2015).

In short, neither the particle nor the resulting PVs can be easily defined. The most characteristic property of PVs is that they are separable and, therefore are different from prefix verbs. The process of combining particles with BVs is very productive. In addition, PVs are highly ambiguous since both constituents can introduce ambiguity. Otherwise, either or both of the constituents can fail to make a contribution to the meaning of the whole. Hence, the resulting PV is not necessarily predictable or transparent. PVs can be seen as instances of so called multiword expressions, which are roughly defined as multiple words having surprising properties which are not predicted by their component words (e.g., “hot dog”).

In the case of PVs, the meaning of the composition can be located on a continuum between transparent and very opaque or idiosyncratic. Interestingly, the construction can trigger a meaning shift. Here, the resulting PV may have an additional meaning that is not applicable to its BV.

These properties make PVs an especially interesting and challenging phenomena for computational models within NLP. We are now going to describe the particle-verb construction and its phenomena in more detail.

2.1.1 Phenomena

What is a Particle?

The most well-known particles are related to prepositions. These are also the most common ones. Such particles may semantically contribute directional *(auf$_{\uparrow}$steigen, to soar)*, locative *(abfliegen, to depart)*, resultative *(aufmachen, to open)*, temporal *(nachsingen, repeat a song)* or aspectual[1] *(anlesen, to read up on sth.)* meaning. Spatial and directional particles are assumed to be most frequent and also most transparent. Furthermore, spatial directional particles provide the diachronic source of other non-spatial particle meanings (Olsen, 1995; McIntyre, 2002). Then again, there are cases where the particle contributes nothing to the complex verb meaning. Here, the verb particle combination may have an idiomatic meaning such as "aufhören" *(to cease*, literally *hear up)*.

Lüdeling (1999) distinguishes between three different views on particles:

1. A particle may belong to any major syntactic category. This is the most generous view on particles. This view is applied by Stiebels and Wunderlich (1994); Booij (1990). Hence, this definition would also allow for nominal particles, such as "klavierspielen" *(piano+play)*, "Rad fahren" *(ride a bike)* or verbal particles such as "spazierengehen" *(to stroll,* literally a combination of *stroll+go)*.

2. The second view assumes that particles are intransitive prepositions (Emonds, 1972; Neeleman and Weerman, 1993; den Dikken, 1995; Zeller, 1997) and although not discussed in detail, the examples given in Eichinger (2000) are all prepositions. In this view, the particles belong to the class of prepositions only. In addition prepositions can be intransitive, like verbs can be intransitive.

3. Lüdeling calls the third view on particles the *P-and-A-particle position*. This view is assumed to be the one that is implicit in most of the work on PVs. Here, particles are assumed to have developed from prepositions, adverbs (davon,

[1] While Slavic prefixes can interact with the grammatical aspect, we refer here to the lexical aspect or aktionsart (Filip, 2012)

> wieder, hin, her ...) or adjectives (offen, klar, fest, frei, ...). This view was adopted by most German grammarians (Paul, 1920; Henzen, 1965) as well as others (Olsen, 1986; Lebeth, 1992; Fleischer and Barz, 2012).

Not all the views on particles are captured by this classification, e.g., Poitou (2003) makes no differences between prefix and PV and, therefore, treats non-separable prefixes (be, ent-, ver-) as particles.

The main focus in this thesis will be on prepositional particles. For most of the experiments in this work, we rely on the following 10 particle types: an *(on, at)*, aus *(from, out of)*, auf *(on, at)*, ein *(in, into)*, ab *(off, from)*, vor *(before, in front)*, durch *(across)*, nach *(after, to, on)*, unter *(below, under)* and über *(over)*. Note, that these particles are also the most common ones[2]. In addition they are highly ambiguous. For example Kliche (2009) distinguishes 18 different semantic meaning contributions for the particle "ab"; similarly, Kempcke (1965) defines 6 main- and 34 subgroups for "an".

To illustrate the meaning contribution of the particle we consider the example of 'grillen' *(to grill)* combined with the particle "an". The resulting PV "angrillen" has at least two possible readings due to the particle i) a partitive reading *(to start grilling something)* and ii) a reading where 'an' marks the beginning of an event *(to start the grilling season)*.

As pointed out by Fleischer and Barz (2012), the various different contributions of a particle can in an extreme case even result in a PV with antonymous senses. Such examples include "abdecken", which can be used to express both *to cover* and *to uncover*, or "auflöten" where something is either *opened* or *closed* via soldering (*löten - to solder*).

Hence, the particle translations provided above exhibit a subset of the possible particle meanings only. The differences in meaning between particles and prepositions are often explained by assuming particles are homonymous[3] with prepositions.

While most particles consist of a single particle, as extensively studied by McIntyre (2001), there are also double particles. Examples include "hineinbauen" *(hin+ein +bauen, to build sth. in sth)'* and "herausziehen" *(her+aus+ziehen, to drag sth. out)*. According to Dewell (2011), double particles contribute a strong directional or spatial meaning and tend to be less lexicalized.

[2] Section 3.5.2 contains a corpus study.

[3] More information on polysemy and homonymy will be provided in the Section 2.2 on word senses and ambiguity.

Productivity

Almost all PVs rely on a verb as a base. Seldom can prepositional particles build PVs using a noun as a base, as in "ausufern" *(aus+Ufer/riverside*$_{\textbf{Noun}}$*, to escalate)*, 'absahnen' *(ab+Sahne/cream*$_{\textbf{Noun}}$*, to rake)*, "anhimmeln" *(an+Himmel/sky*$_{\textbf{Noun}}$*, to adore sb.)*, or an adjective, as in "ausdünnen" *(aus+dünn/thin*$_{\textbf{Adj}}$*, to thin out)* and "aufheitern" *(auf+heiter/cheerful* $_{\textbf{Adj}}$*, to cheer sb.)*[4].

Almost every verb can be combined with a particle, hence the construction of PVs is a very productive phenomena. Subsequently, a verb can also be combined with numerous different particles, this can be even done to create a stylistic effect as illustrated by the following two lines taken from a recent German music song:[5]

Du hast mich an+gezogen, aus+gezogen, groß+gezogen und wir sind um+gezogen [...]
You dressed me, undressed me, raised me and we moved away, [..]

According to Fleischer and Barz (2012), the combination particle (and prefixes) represents the most important common tool for the systematic creation of novel word forms. This productivity is restricted to prepositional and adverbial particles; thus, verbs or nouns as base are significantly less productive (Stiebels, 1996).

In addition, the literature agrees on the observation that PVs have a systematic idiomaticity of some kind. Still, there is no agreed view on word formation and the underlying productive schema that is used to create new PVs (neologisms). Moreover, the distinctions between the mechanisms of word formations are often not clearly defined. It is often assumed that neologisms are created by applying an abstract global rule-based productive schema or by performing a more local analogy that depends on a concrete lexical target. Another word formation mechanism, briefly mentioned in McIntyre (2002) but discussed in detailed in Gerdes (2012), is blending. Here, new PVs are created by substituting either the BV or the PV. More importantly, it should be mentioned that these views are not necessarily different. Gerdes (2012) describes them as a continuum and shows that neologisms can usually be seen as the result of both perspectives.

Regarding semantic change and more general diachronic semantics, PVs and prefix verbs are often explicitly left aside. Harm (2000) argues that PVs are fraught with problems. Conventionally, words in natural language adapt to new meanings, rearrange current meanings, or lose old meaning. However, new PVss can be coined

[4] Some examples are taken from Fleischer and Barz (2012)
[5] AnnenMayKantereit - *Oft gefragt* (2016)

based on another morphological reinterpretation or possible reading. Hence a new PV may be created that is not related or derived from the same preexisting form (same sequence of characters). Geeraerts (1997) refers to this phenomenon with the term "morphological polygenesis".

We would like to point out three different studies, that illustrate the potential of productivity and the creation of novel PVs. Felfe (2012) conducted an interesting experiment, in which 19/20 German native speakers confirmed that they did not know the PV "anschlafen" (*an+schlafen, to start sleeping)* when they were presented with sentences from a corpus, yet all of them could understand the meaning without problems. In a similar vein, Springorum et al. (2013a) systematically created novel particle-verb neologisms. The subjects were perfectly able to associate a meaning to these verbs and to construct example sentences for them. In addition, different subjects agreed to a large degree on the semantic meaning they attributed to the newly formed lexical items. Gerdes (2012) conducted a manual analysis based on a collection of press release texts with respect to occurrences of "an-" and "auf-" PVs. According to him, approximately 45% of all "an" and 50% of all "auf-" PVs found in a corpus were not listed in a German lexicon.

Particle Verb Meaning Shifts

We have already seen that the particle is highly ambiguous. On the other hand, the BV can exhibit an unpredictable behavior, as well.

Numerous verbs are ambiguous and keep this ambiguity when in combination with a particle. The verb "strahlen" means *to beam/shine* or *to smile*. When combined with the directional meaning of "an" the resulting PV "anstrahlen" can either refer to *beam at something* or *smile at somebody*.

Furthermore, there are cases where the contribution of the particle is predictable, but the semantics of the BV is different. Dehé et al. (2002) mention the verb "eintrudeln" *(to arrive in dribs and drabs)* as an example. This PV entails the spatial contribution of "ein"; however the verb "trudeln" *(to spin)* exhibits different semantics in this combination. Analogously, the lexicalized PV "überbraten" *(to whack sb. over the head with sth.)* entails the contribution (over the head) from an image related to the "über" particle, whereas the BV means *to fry*. For both examples BVs, "braten" and "trudeln", the observed meaning is not observed with any other particle combination.

There are also constructions in which the directionality of the particle contradicts that of the BV (McIntyre (2002) calls them *pseudoreversatives*), and the resulting compound is in some way the antonym or opposite to its BV; examples include “auseinander+montieren” *(to disassemble)*, “ab+schwellen” *(to detumesce)* or “los+binden” (*to untie*).

Some German verbs do not even exist on their own, but only in combination with particles, e.g., there is no verb “brezeln” *(to pretzel?)* but a PV “aufbrezeln” *(to get dressed up)*.

Semantic analyses, e.g., Lechler and Roßdeutscher (2009a) (for “auf”), Kliche (2009) (for “ab”), Springorum (2009) (for “an”) and Haselbach (2011) (for “nach”) demonstrate that each particle has several different readings that form regular patterns depending on the context. The majority of particle-verb constructions represents such compositional combinations and can be explained by patterns.

In the same way, there are some BVs that seem to behave remarkably similarly with respect to meaning shifts. For example “brummen” and “donnern” undergo a similar shift from literal to non-literal when combined with the particle “auf”. The resulting PV is ‘aufbrummen” with one of its meaning being ‘jemandem eine Aufgabe aufbrummen” *(to forcefully assign a task to someone)*. This PV is constructed using the sound verb “brummen” *(to hum)*, which has nothing in common with the previous mentioned semantics of “aufbrummen”. Interestingly, a similar shifted meaning can be found for another sound verb, namely ‘donnern” *(to rumble, to thunder)*. Both base verbs describe a displeasing loud sound. The resulting PVss “aufbrummen” and “aufdonnern” are near synonyms and share the same non-literal meaning. Typically, not all senses of a PV undergo meaning shifts, both verbs can also be used in the literal sense as in “der Motor donnerte/brummte laut auf” *(the engine started to roar)*. Such regularities can be found across a variety of PV combinations, for example “zischen” *(to hiss)*, “dampfen” *(to steam)*, “rauschen” *(to whoosh)* and “brausen” (*to swoosh)* are clearly semantically related and when combined with ab, they all share the shifted meaning of leaving or disappearing *(to vamoose)*. Analogous “aufspruden” *(bubble up)*, ‘aufkochen” *(to boil up)* and “aufbrausen” *(flare up)* all share the meaning shifted sense of “become angry”.

Furthermore, Springorum et al. (2013b) provide a corpus-based case study on regular meaning shift conditions for German PVs. They argue that there are regular mechanisms in meaning shifts of a BV in combination with a particle. Hence, it is likely that such meaning shifts apply across a semantically coherent set of verbs. Additionally, new verb constructions can be created either by direct analogy or by applying an abstract productive rule-based schema.

Where are Particle Verbs? Syntax or Morphology

There is no agreed definition of PVs. This is because PVs share properties from both, morphological objects and syntactic constructions. Hence, the question is not really *what are particle verbs?*, but rather *where are particle verbs?* or how to draw the boundaries. Lüdeling (1999) assumes the following definition is one on which everyone agrees:

> [...] *"particle verbs are constructions that consist of a verb and a preverb and that behave like words in some respects and like syntactic constructions in others. [...] "*

Thus, PVs in German (and also in Dutch) possess properties of words and syntactic phrases. This observation led to the ongoing debate about whether PVs are instances of words (morphological objects) or syntactic combinations. We want to illustrate this behavior by looking at the examples taken from Zeller (2001b):

a) "weil er sich dem Gegner [unterwirft]." (**prefix verb**)
because he surrenders to the enemy

b) "weil er ihm seine Verfehlungen [vorwirft]." (**particle verb**)
because he reproaches him with his lapses

c) "weil er ihm den Brief [in den Briefkasten wirft]." (**phrasal construction**)
because he throws the letter into his letterbox

Zeller argues that at first glance, the PV in b) seems to behave like the prefix verb in a). Both constructions share typical word-like properties, for example both verbs *unterwerfen* and *vorwerfen* are non-transparent. Non-transparency or semantic idiosyncrasy is a property of words that have a meaning that diverges from the combined contribution of their constituent parts. In this case, their meaning is not based on the literal meaning of *werfen*.

On the other hand, the phrasal construction in c) is highly transparent. In addition, PVs and prefix verbs can be used as input for the morphological rule that derives a noun, given a verb. While one can derive the noun *Unterwerfung* from a) and similarly the noun *Vorwurf* from b) it is not possible to derive a noun from the phrasal construction in c).

All these arguments make valid points for analyzing PVs as morphological objects. On the other hand, it can be seen in a.2) that the whole prefix verb has undergone a movement to the left. However, the PV in b.2) does not behave like the prefix verb; here, only the BV is moved. Thus, the behavior of the PV is more similar to the example given in c.2), where only the main verb is moved and the prepositional phrase remains.

a.2) "Er [unterwirft] sich dem Gegner." (**prefix verb**)
He surrenders to the enemy

b.2) "Er **wirft** ihm seine Verfehlung [vor]." (**particle verb**)
He reproaches him with his lapses

c.2) "Er **wirft** ihm den Brief [in den Briefkasten]." (**phrasal construction**)
He throws the letter into his letterbox

Hence, the literature on German PVs contains different views on this phenomenon. The presented division here is based on the commonly cited literature and should not be seen as a complete overview of the literature. There is rich literature that argues for the syntactic view (Riemsdijk, 1978; Groos, 1989; Zeller, 1997; Lüdeling, 1999; Zeller, 2001b,a; Müller, 2002). In contrast, there is also large literature treating PVs as morphological elements (Booij, 1990; Neeleman and Weerman, 1993; Neeleman and Schipper, 1993; Stiebels and Wunderlich, 1994; Stiebels, 1996; Olsen, 1997). Further, approaches on particle-verb structure can be divided into more fine-grained views. For example, the syntactic view can be divided into views where the particle and the direct object form a constituent, or views where the verb and particle form a constituent.

Separability and Prefix Verbs

The best-known characteristics of PVs are their syntactic separability. PVs may appear together as one word, as in sentence b), or may appear syntactically separated, as sentence b.2) shows. Separated PVs can be challenging for NLP applications, such as machine translation or parsing. The potential number of intervening words between the BV and the particle can be very large. An exaggerated illustration of such distances can be seen in the example from Twain (1880):

> "Die Koffer waren gepackt, und er **reiste**, nachdem er seine Mutter und seine Schwestern geküsst und noch ein letztes Mal sein angebetetes Gretchen an sich

> gedrückt hatte, das, in schlichten weißen Musselin gekleidet und mit einer einzelnen Nachthyazinthe im üppigen braunen Haar, kraftlos die Treppe herabgetaumelt war, immer noch blass von dem Entsetzen und der Aufregung des vorangegangenen Abends, aber voller Sehnsucht, ihren armen schmerzenden Kopf noch einmal an die Brust des Mannes zu legen, den sie mehr als ihr eigenes Leben liebte, **ab**."
>
> *"The trunks being now ready, he* **de-** *after kissing his mother and sisters, and once more pressing to his bosom his adored Gretchen, who, dressed in simple white muslin, with a single tuberose in the ample folds of her rich brown hair, had tottered feebly down the stairs, still pale from the terror and excitement of the past evening, but longing to lay her poor aching head yet once again upon the breast of him whom she loved more dearly than life itself,* **parted**."

As mentioned already, most particles are homonymous or polysemous to prepositions. Thus, an automatic system might misinterpret the particle and fail to recognize longer syntactic dependencies. Additionally, Volk et al. (2016) report that frequent MWEs[6] containing adverbs or prepositions, such as "ab und zu", "auf und ab", "durch und durch", "nach und nach", "nach wie vor" lead to false part-of-speech tags.

The literature, particularly Dewell (2011) and partially Khvtisavrishvili et al. (2015), provide detailed information on the phenomenon of particle-verb separability. PVs are syntactically separated from their BVs in main clauses, usually occurring in a final position in the clause (as in sentence e)). They can also appear separated in interrogative clauses (questions) and imperative clauses (commands). They occur directly attached to the front of the BV only in the simple infinitive (sentence d)), in participle perfect (mit**ge**kommen) or in subordinate clauses when the BV is placed in final clause position (sentence f))[7]. Hence, the syntactic separability depends on the type of clause and the status of the BV (finite/infinite).

d) "Möchtest du *mitkommen*?"
Would you like to come with us?

e) "Ich *komme* gerne *mit*."
I would like to come with you

f) "Ich bin jedem dankbar, der *mithilft* und meinen Fragebogen *ausfüllt*."
I am grateful to everyone who helps and fills out my questionnaire.

[6] They call them bi-particle adverbs

[7] Examples d), e) and f) are taken from Dewell (2011)

Unlike prefix verbs, only PVs are separable. In addition, both types of verbs differ with respect to the stress pattern. For prefix verbs, the verb root usually receives the stress (entkommen). Particles are prosodically strong, hence the primary stress falls on the particle (mitkommen) (Biskup, 2011). Prefix verbs on the other hand are never separated. Common prefixes of German include *be*, *ent*, *er*, *hinter*, *miss*, *ver* as well as *zer*.

Some prefix verbs are also preposition related and can be mistaken for PVs; a particularly interesting example is illustrated in "umfahren" used as a prefix verb in Figure 2.1 and as a PV in Figure 2.2.[8] Dependent on its usage, the two senses are antonyms, and therefore especially difficult for German learners.

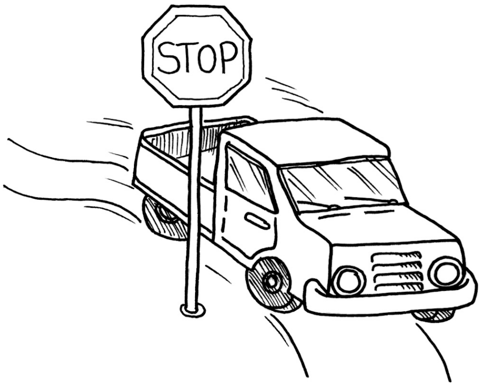


Figure 2.1: "Er umfährt das Schild." (**prefix verb**) *He drives around the sign.*

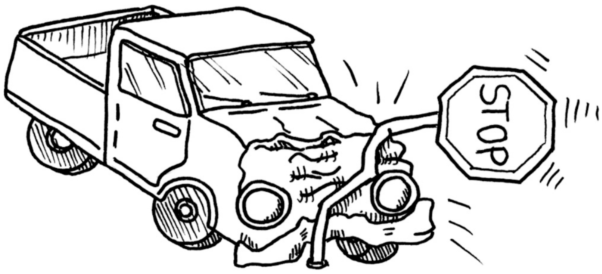


Figure 2.2: "Er fährt das Schild um." (**particle verb**) *He drives over the sign.*

Prefixes that can also be particles include *durch*, *über*, *um* and *unter*. The literature, particularly the pedagogical grammar from Helbig and Buscha (1998) contains hints that prefix verbs are even more likely to be lexicalized and used figuratively. On the other hand, Dewell (2011) shows clearly that this tendency has too many exceptions. Still, distinguishing prefix verbs from PVs is particularly challenging for German learners. It requires a feeling of these patterns to construct new verbs that other speakers can understand and to know when to use one and when to use the other.

2.1.2 Research on Particle Verbs

PVs represent an interesting phenomenon across languages and offer many research directions. Hence, the literature on this subject is comparably rich. To illustrate, the

[8] Both figures are taken from the website, Learn German on Lingolia (2018).

outdated but extensive bibliography from the research project *Particle Verb Formation in German and English* contains ≈230 entries[9]. Additionally, entire monographs have been written focusing on a single particle. Hence, this section aims to provide a rough overview of the different research areas, with a particular focus on computational work.

Theoretical: Until now, most work on German PVs has been devoted to theoretical investigations that have provided mainly structural considerations regarding morphological or syntactic properties. A less scientific and more general view with respect to the phenomenon of PVs can be found in the books on German word formation by Eichinger (2000) and Fleischer and Barz (2012).

Stiebels (1996) analyzes complex verbs, which are based on three particles and three prefixes. Her approach treats prefixes and PVs similarly and can be seen as a strong lexical approach within the framework of lexical decomposition grammar. Here, PVs are morphological objects and new PVs are formed in the word formation component of grammar. This is in contrast to the comprehensive study by Lüdeling (2001). Lüdeling argues for a syntactic view, treating PVs as phrases. Lüdeling's work is particularly influential because she argues that there is no clearly defined class of PVs. According to her, PVs possess no distinct properties that would define their own class. Hence, she argues that there are no PVs.

More semantically motivated is the work from McIntyre (2002). His study shows the wide range of PV phenomena across English and German. Furthermore, he shows that meanings of many seemingly idiosyncratic PVs can be explained by composition. Felfe (2012) represents a detailed analysis for the particle an-. Felfe employs frame semantics to explain the compositionality of the verb and PV construction.

Although less scientifically focused, the book from Dewell (2011) includes a detailed descriptive survey and extensive examples on the differences between prefix and particle-verb usage. Focusing on "durch-", "über" and "unter-", he illustrates that the separate prefix and respective particle carry already meaning and reveal patterns in the German verb system. More recently, Gerdes (2012) conducted a large corpus study focusing on "auf-" and "an-" PVs. His work focuses in particular on infrequent novel neologisms.

PVs are also of interest in psycholinguistics. They have been studied with respect

[9] Based on December 2017 http://ling.uni-konstanz.de/pages/home/dehe/bibl/PV.html

to language processing and acquisition (Svenonius, 1996; McIntyre, 2002). Richter (2010) looked at the different errors that children make with respect to particle and prefix verbs. Her findings show that children distinguish already between prefix and PV. Lüdeling and De Jong (2002) conducted a priming experiment, looking at the reaction time of participants with respect to the degree of PV transparency. Her results show no difference with respect to the reaction time of opaque and transparent PVs leading to the conclusion that PVs have their own (phrase-like) status in the mental lexicon. Frassinelli et al. (2017) conducted a lexical decision experiment to investigate the directionality of the particles "an" and "auf". They hypothesize that "an" is primarily associated with a horizontal directionality while "auf" is associated with a vertical directionality. They systematically created mismatches between particle and BV (e.g., "auf" with a horizontal BV) and report that it takes significantly longer to process such mismatching PVs.

Computational: Given the large amount of theoretical work on (German) PVs, the literature on computational models is comparably small. While compounds have been a recurrent focus of attention within computational linguistics, research on PVs has played a comparably marginal role among this.

Regarding German PVs, most of the work focuses on their identification and compositionality. Earlier work includes the work on subcategorization by Schulte im Walde (2006) and Hartmann (2008). Hartmann models subcategorization transfer between BV and PV to strengthen PV-BV similarity using a distributional model. In addition, she introduces the first gold standard containing compositionality judgments for 99 German BV-PV constructions.

In a similar vein Bott and Schulte im Walde (2014a) predict syntactic slot correspondences between syntactic slots of base and particle verb pairs. Their automatic method obtains a fair degree of success in a classification setup by relying on a slot specific vector space model and cosine similarity.

Kühner and Schulte im Walde (2010) apply a soft-clustering approach to determine the compositionality between the BV and the PV. They assume that compositionality correlates to cluster membership and they evaluate their approach on the data from Hartmann (2008).

Bott and Schulte im Walde (2014b) explore various distributional models, varying window size, contextual parts of speech and feature weighting to predict compositionality. In addition, they propose a reconstruction of PV lemmata in cases where the parser outputs the BV (separated from the particle verb). Their findings show that a purely windows-based approach can perform well on this setup. Moreover, the reconstruction of separated PVs tends to increase performance.

Bott and Schulte im Walde (2015) investigate generalization methods, such as topic models, GermaNet (a large lexical database created by Hamp and Feldweg (1997)) and singular-value decomposition to enhance compositionality predictions. None of their methods obtains superior performance to their previous model (a standard bag-of-words model).

To enhance the research on the compositionality of German PVs, Bott et al. (2016) introduced a novel and gold standard resource. Their collection contains 400 PVs, balanced across several particle types and three frequency bands, and accompanied by human ratings, created by manually rating the degree of semantic compositionality.

Another line of research categorized particle meanings by relating formal semantic definitions to automatic classifications (Rüd, 2012; Springorum et al., 2012). To the best of our knowledge, there is no computational work on assigning senses to particles[10].

Among other derivational patterns, “über”, “an” and “durch” PVs have also been included in work that aims to study the semantic behavior of derivational processes (Kisselew et al., 2015; Lapesa et al., 2017).

There is also work that focuses on the detection of German PVs and, in particular, on the reconstruction of separable PVs. Volk et al. (2016) designed an algorithm to detect separated PVs for large-scale corpus annotation. Their method attaches the particle to the nearest preceding finite verb if the word passes a precompiled lookup list. Different adaptions of Volks method were applied on the corpus of spoken German in Batinić and Schmidt (2017). Both approaches focus on the creation of a reliable list of possible PVs to enhance PV detection.

Nießen and Ney (2000) propose to concatenate separated PVs to reduce the number of out-of-vocabulary terms for machine translation. Other approaches in the context of machine translation detect PVs and other MWEs with parallel data (Fritzinger, 2010). Schottmüller and Nivre (2014) systematically explored the impact of PVs in a German to English setup. They used two state of the art machine learning systems[11]

[10] Although we did not model particles senses, we conducted a type based (soft) classification on preposition senses (Köper and Schulte im Walde, 2016), which can be seen as a related task.

[11] The systems were Google Translate and Bing Translator. The systems relied on statistical methods (SMT) in 2014. Hence, this was before the usage of Neural Machine Translation (NMT) in 2016.

and evaluated all the translations manually. Their findings show that the quality decreases when translating sentences that contain PVs. Moreover, they suggest that PVs can be replaced with a synonymous BV (simplex verb). In more detail, only 71.6% of all PV translations were correct in contrast to 90.7% of the simplex translations. Their analysis reveals that, in many cases, the wrong translations is due to the separated particle since the system tends to translate the BV correctly.

PVs are even challenging for lexical-semantic databases. For example, Hoppermann and Hinrichs (2014) define the criteria to integrate PVs into GermaNet. They distinguish between compositional and non-compositional by taking the semantic relation into account. The underlying assumption is that there is always either a conceptual (hyper/hyponym) or lexical (synonym/antonym) relation between PV and the respective BV, e.g., laden *(to load)* is a hypernym of aufladen *(load up, to charge)*.

Another line of computational research focused on the task of synonym extraction. This has been done using distributional similarity and parallel data (Wittmann et al., 2014), graph clustering (Wittmann et al., 2016) and our recent approach with soft-clustering (Wittmann et al., 2017). Khvtisavrishvili et al. (2015) present a large-scale empirical corpus study on separability. They report high variation in the frequencies with which PVs occur in different syntactic paradigms. However, they could not provide a definitive answer as to which factors determine this behavior.

Finally, our own publications, described in more detail in this thesis, cover multiple research directions. We systematically investigate compositionality using distributional representation and visual representation (Köper and Schulte im Walde, 2017b). Additionally, we address the problem of (PV) vector prediction, inspired by work that models derivation using distributional semantics (Köper et al., 2016). Furthermore, we perform a token based literal vs. non-literal classification (Köper and Schulte im Walde, 2016).

In another line of research we investigate the usage of sense specific vectors and evaluate these representation with respect to compositionality, semantic classification and the detection of non-literal language (Köper and Schulte im Walde, 2017a). Finally we conduct a type-based classification on BV-PV analogies to investigate the phenomenon of regular meaning shifts (Köper and Schulte im Walde, 2018).

Our most recent work presents a collection of sentences to assess meaning compo-

According to the comparison between phrase based SMT and NMT from Popović (2017) it was found that NMT systems are doing better for verbs and separated compounds. Hence, it is likely that machine translation for PVs became better after 2016.

nents. The sentences in this collection were annotated for non-literalness, directionality, as well as source (for BVs) or target (PVs) domains (Schulte im Walde et al., 2018).

Across Languages: With respect to approaches across languages, there is a considerable amount of work on English PVs. Here, the work includes automatic extraction or identification of PV constructions from corpora (Baldwin and Villavicencio, 2002; Li et al., 2003; Baldwin, 2005; Villavicencio, 2005; Kim and Baldwin, 2006) and more recently Nagy and Vincze (2014). The correct detection of PVs can be useful for other tasks, as shown by Constable and Curran (2009). They report an increase in the F-score when particle-verb information is integrated into a parsing system.

A considerable amount of work has also been done on the determination of compositionality. Notably, the first approach was conducted by McCarthy et al. (2003). They exploited various statistical measures, such as vocabulary overlap and nearest neighbors to predict the degree of compositionality. Baldwin et al. (2003) adjusted Latent Semantic Analysis (LSA) models for English PVs and their constituents to determine compositionality. Similarly, Bannard et al. (2003) experimented with four corpus-based approaches (context space models). Bannard (2005) defined the compositionality of an English PV as an entailment relationship between the PV and its constituents. The assumption here is that lexical contexts for a PV will be more similar to those of a given component word if that component word is contributing its simplex meaning to the phrase. For example "put up" entails "put". The evaluation was conducted by comparing a distributional model against human entailment judgments. All these approaches were type-based, and predicting the compositionality was mainly concerned with PV–BV similarity, not taking the contribution of the particle into account. There is large work on modeling preposition senses across languages (Litkowski and Hargraves, 2005, 2007; Köper and Schulte im Walde, 2016). However, in cases where the particle semantics was respected, such as Bannard (2005), the results were disappointing because modeling particle senses is still an unsolved problem.

Cook and Stevenson (2006) conducted a token based classification for the English particle "up". They compare word co-occurrence and linguistically motivated syntactic slots and particle-specific feature dimensions. Their results show that best performance across the datasets is obtained using all the linguistic features. In a similar vein, Bhatia et al. (2017) applied a heuristic, relying on the WordNet (Fellbaum et al., 1998) hierarchy, to classify compositional vs. non-compositional usage of multiple English particles, based on their appearance in a sentence.

The literature contains very little work on languages other than German or English. According to Dehé (2015), most other Germanic languages share the interesting phenomenon of PVs, including languages in which the particle precedes the verb in the infinitive, such as Dutch and low-resource languages such as Yiddish and Afrikaans. Interestingly, the same computational research directions have also been explored for Estonian PVs, namely automatic PV extraction (Kaalep and Muischnek, 2002; Aedmaa, 2014), compositionality (Aedmaa, 2017; Muischnek et al., 2013) and the manual annotation of large corpora (Kaalep and Muischnek, 2006, 2008). In addition there is our recent work, on the automatic detection of non-literal language for Estonian PVs (Aedmaa et al., 2018).

Other work includes the large scale corpus study on Hungarian PVs by Kalivoda (2017). Here, the focus relies on the distance between the particle and BV, as well as on the factors that determine whether a particle should stay close to its verb.

2.2 Ambiguity and Sense Discrimination

> “ *One morning, I shot an elephant in my pajamas. How he got in my pajamas, I don't know.*
> Groucho Marx, *Animal Crackers*, 1930 ”

Ambiguity means that something can be understood in at least two ways. The existence of ambiguity has no trivial explanation, as discussed in Wasow et al. (2005). Languages are not exclusively built for precision. It is often assumed that ambiguity arises from the need that a language should be efficient from the perspective of the speaker (Zipf, 1949).

However, ambiguity is one of the main reasons why language processing is difficult. According to Manning and Schütze (1999) a system needs to know at least *“Who did what to whom?”*. Language is clearly ambiguous in multiple forms. Although our focus relies on lexical ambiguity, we will use this section to provide a short overview of the various linguistic forms of ambiguity and clarify important terminology with respect to ambiguity and word senses.

The Groucho Marx joke, quoted above, represents an example of **structural ambiguity**. The prepositional phrase *in my pajamas* can either modify the direct object (the elephant) or it can be attached to the verb (and therefore also the shooter) as Figure 2.3 illustrates. Structural (or syntactic) ambiguity arises whenever two or more possible syntactic structures cause multiple interpretations. The computational approach to this problem is syntactic parsing, where a structure in the form of a tree is assigned to a given input string (sentence).

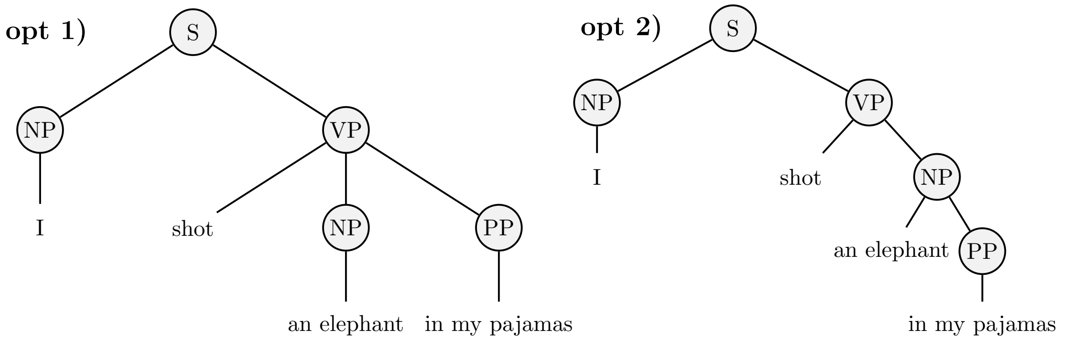


Figure 2.3: Two simplified parse trees for the sentence *"I shot an elephant in my pajamas"*. Each tree results in a different semantic interpretation.

A subcategory of structural ambiguity is called **scope ambiguity**, discussed in detail by Chierchia and McConnell-Ginet (2000). Here, the ambiguity arises when two or more quantifiers or a negation take scope over each other, as the following example illustrates:

a) *"Every farmer loves a donkey"*

The sentence has two readings i) a single donkey is loved by every farmer or ii) for every farmer there exist a donkey, such that the farmer loves the donkey.

While syntactic ambiguity is concerned with the structure of sequences of words, **lexical ambiguity** deals with multiple interpretations of a single word. In lexical ambiguity (sometimes semantic ambiguity or homonymy) the ambiguity resides in the word, as in the most classic English example *bank*, which can either refer to a financial institution or the edge of a river. While both meanings are clearly unrelated, the lexical items have the same form. In fact, the literature often makes a clear distinction between related and unrelated ambiguous words. Completely unrelated words are considered **homonymous** word pairs, such as the verb vs. noun senses of *bear* or the fish vs. instrument sense of *bass*. On the other hand, related but different meanings are called **polysemous**, with the antonym **monosemous** meaning a word having only one interpretation. A nice example for two polysemous senses is given by Akmajian et al. (2001):

"Sports Illustrated can be bought for 1 dollar or 35 million dollars."

The sense related to a single magazine, and the one related to the entire company are clearly connected but nevertheless represent two different meanings. In a similar vein, one could even divide cases of the typical clear-cut homonym *bank* as polysemous e.g., Pustejovsky (1995) mentions the example of the abstract financial institution (*The bank raised its interest rates.*) in contrast to the concrete physical building (*John walked into the bank*). Other examples of polysemy are:

b) *to get* get sick, get a raise, get angry, get it (understand)

c) *wood* material, geographical area with trees

Polysemy is a highly frequent phenomenon and a lot of work has been devoted to its treatment (Nunberg, 1992; Pustejovsky, 1995; Copestake and Briscoe, 1995). Often, a word gains new usages over time (semantic change) (Murphy, 2010); hence, polysemous words share their etymological background and, therefore, belong to the same semantic field.

In the context of PVs, most studies across language attempt to unify various senses treating PVs, and particularly the contribution of the particle, either as polysemous or monosemous rather than homonyms (Lindner, 1983; Lieber and Baayen, 1993; McIntyre, 2001). Stiebels (1996), however, represents a position that does not relate the various uses of a particle, this view is criticized for lacking generalization in Dehé et al. (2002).

However, it is not always obvious if a sense was created due to semantic change or if the words just accidentally have the same form. Hence, "*the distinction between homonymy and polysemy is notoriously elusive*" (as cited in Lipka 1975). Even one of the most commonly used resource in computational linguistics, WordNet (Fellbaum et al., 1998), provides only a word sense inventory without distinguishing between the two. Instead of relying on etymology, we adapt a context-centric definition of lexical ambiguity, in line with Firth (1953) "*the complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously*". Therefore, we apply the same definition as in Depraetere and Salkie (2017), which is, "*When items have multiple meanings which are mutually exclusive in every context, we shall call this lexical ambiguity, be this homonymy or polysemy*". In other words, we regard lexical ambiguity/homonymy and polysemy as synonymous. Consequently, lexical ambiguity can only be resolved by looking at the context of a lexically ambiguous word.

The task of identifying the meaning of a word in context is called Word Sense Disambiguation (WSD). WSD is one of the oldest problems in computational semantics and was already introduced as a separate task in the early stage of machine translation in the 1940s.

WSD usually relies on a set of possible senses for a given word (a sense inventory). The term word sense discrimination is used to refer to computational models that work without the use of external resources. Such models perform automatic sense identification and can produce a sense inventory. Hence, word sense discrimination is used to refer to unsupervised approaches that operate directly on raw unannotated corpora.

Linguistic forms of ambiguity contain even more different types, such as phonetic ambiguity, referential ambiguity or pragmatic ambiguity. However, these types are beyond the scope of this work. Another essential point is that the process of composition will create many different possible readings if a sentence contains multiple sources of ambiguity. Therefore, a NLP system needs to determine the right syntactic structure, word category, semantic scope, and word sense.

2.3 Non-Literal Language

Non-literal or figurative language describes language that goes beyond the literal meaning of words or phrases. More importantly, non-literal language appeals to one's imagination and can be used to explain new ideas or project complex meanings. There are multiple forms of non-literal language, such as a simile (comparison), personification (giving human qualities to objects), metaphor (detailed in Section 2.3.1), idioms (detailed in Section 2.3.2), or hyperbole (extreme exaggeration).

We apply here a view on the phenomenon, that assumes there is a continuum, ranging from very literal to non-literal language. Thus each type of non-literal language usage can be seen as one point on this continuum. Here, idioms can be considered as one extremes at the end of a continuum, as opposed to very literal language.

Moreover it is highly frequent in language and represents one of the most difficult tasks regarding NLP and, more generally, AI. A listener is only able to unpack the meaning of an utterance if they share conceptual structures and knowledge. To illustrate this, consider the following two example sentences:

a) "Die Demokraten wollen alte Strukturen **aufbrechen**."

The democrats want to upset old structures.

b) "Das kann nicht sein Ernst sein, ich glaube, er wollte dir einen Bären **aufbinden**."
This cant be serious, I think he wanted to trick you. (literal wanted to tie a bear on you)

While the most frequent sense of *aufbrechen* is related to a physical action of breaking something e.g., to break a door, the actual meaning in a) can only be understood if a reader knows that the intended action, as well as the mentioned structures, are abstract. Similarly, example b) would literally mean to ''tie a bear on someone". The example illustrates further that in non-literal language the meaning of the whole expression cannot be constructed from the meanings of the parts.

While computational models are able to store and process huge amounts of information, they struggle with the performance of such high-level semantic tasks that humans solve effortlessly. We are now going to provide theoretical background for the most important forms of non-literal language with respect to German PVs, namely metaphors, as in the previous example a) and idioms, as seen in example b).

2.3.1 (Conceptual) Metaphors

A metaphor arises when one domain is explained in terms of another conceptual domain. Humans possess the ability to reason via analogy; as a result of this process, metaphors arise (Gentner et al., 2001). In the past, it was assumed that metaphors are infrequent and their usage was seen as an artistic device more or less limited to poetic use. Lakoff and Johnson (1980), however, show that metaphors are highly frequent in every area of human activity, ranging from the language in our everyday lives to scientific language. Some metaphors are so common that we may not even notice that they are metaphors. Metaphor is a very productive phenomenon that allows humans to create novel expressions. A large-scale corpus analysis even estimates that "on average, one in every seven and a half lexical units is related to metaphor" (Steen et al., 2010). The literature contains different views on metaphors, all of which share the idea of concept mapping (Black, 1962; Wilks, 1975; Gentner, 1983). In this thesis, we focus on the prominent conceptual metaphor theory by Lakoff and Johnson (1980).

The conceptual metaphor theory (Lakoff and Johnson, 1980) distinguishes between two types of domains, a SOURCE domain and a TARGET domain. A metaphor represents a mapping from SOURCE to TARGET, where the concrete SOURCE domain is used to understand the abstract TARGET concept. Furthermore, a metaphor is a cognitive phenomenon and not only a linguistic phenomenon; therefore, the conceptual metaphor is not limited to verbal modality only. A metaphor can also be expressed via visual modality (Forceville, 1994; Weelden et al., 2012). However, a linguistic metaphor or metaphorical expression can be seen as the manifestation of a conceptual metaphor in natural language.

The relation between source and target domains is usually asymmetric and unidirectional from source to target, therefore source and target are not interchangeable. Using the well-known example from Lakoff and Johnson (1980), "love is a journey." we might describe LOVE using the concept of JOURNEY. This is done by mapping the properties of the two domains, e.g., concrete travel destination to abstract relationship goals, travelers to lovers, and physical obstacles to relationship problems resulting in concrete expressions for *love* such as *"We've gotten off track."*, *"We're at a crossroads."* or *"We'll just have to go our separate ways."*. Conversely, the same does not hold true vice-versa as we cannot describe a JOURNEY in terms of LOVE.

There are also studies that claim that metaphors can be bidirectional (Zhong and Liljenquist, 2006; Zhong and Leonardelli, 2008). Furthermore, Dancygier and Sweetser (2014) maintain that even seemingly exceptional cases, such as the PEOPLE↔COMPUTER metaphor, represent unidirectional mappings. In brief, given the observation of expressions such as *"My memory banks are scrambled."* and *"My computer is being stubborn and difficult today."*, one might assume that the mapping between COMPUTERS ARE PEOPLE and PEOPLE ARE COMPUTERS is reversible. However, Dancygier and Sweetser (2014) state that the actual mapping is unidirectional and should rather be labeled as HUMAN COGNITIVE PROCESSING IS COMPUTER INFORMATION PROCESSING and APPARENTLY ERRATIC ASPECTS OF COMPUTER BEHAVIOR ARE EMOTIONAL MOOD-BASED ASPECTS OF HUMAN BEHAVIOR.

While the previous examples were based on the English language, the principle of conceptual metaphors holds for all languages. Although each language may use different domain mappings there is even evidence of universal metaphors, such as the TIME IS SPACE mapping (Alverson, 1994). We are now going to illustrate a metaphor mapping based on the following two German sentences with the PV "aufblühen" (*to flourish*):

c) (lit.) "Die Knospe der Blume **blüht auf**."
*The flower bud is **blooming**.*

d) (non-lit.) "Die Stadt **blüht** durch Industrie und Gewerbe wirtschaftlich **auf**."
*Due to industry and trade the city's economy **thrived**.*

The literal sense of *aufblühen* is exclusively used to refer to plants, as illustrated in sentence c). Therefore the well-known physical concept of PLANTS is used in the metaphorical sense d) to describe the clearly more abstract concept of the economic development of a city. The mapping performed in d) can be seen as mapping from the concrete source domain PLANTS → the abstract target meaning ECONOMY.

It should be mentioned that Kövecses (2002) conducted a manual study based on a great number of English metaphor dictionaries to determine which source concepts and target concepts are most commonly used. Köveces claims that we are most likely relying on concrete source concepts, which we are especially familiar with, such as the HUMAN BODY, ANIMALS, or PLANTS, whereas the highly common abstract target domains contain classes such as EMOTION, DESIRE, ECONOMY or POLITICS. Here our work (Schulte im Walde et al., 2018) represents the first data driven study with respect to source-target mappings for German PVs, as discussed in detail in Section 6.1.

Furthermore, recent work on English metaphors showed that the metaphorical uses of words carry stronger emotions than their literal uses (Mohammad et al., 2016). In a similar vein, our work (Köper and Schulte im Walde, 2018) confirms these findings, as our computational model of PV meaning shifts performs better when relying on emotion information (Section 6.2).

2.3.2 Idiomatic Expressions

Idiomatic expressions or idioms are defined as non-literal statements where the meaning cannot be derived from the meanings of their individual compositional parts (Nattinger and DeCarrico, 1992). This non-compositionality of meaning challenges theories of syntax, since the words of many idioms form no constituents. A famous English example is "kick the bucket" meaning "to die". Unlike lexical metaphors (single words), most idiomatic expressions span over multiple words. Interestingly, many idioms were originally used in the literal sense but lost the literal meaning over time due to semantic change. This happens when the literal expression is no longer used or succumbs to the metaphorical meaning. To illustrate, consider the following two idiomatic expressions, using a German PVs in expression e) and f):

e) "Jemandem das Wasser abgraben"
(lit) *to dig up water*
(non-lit.)*to take away sb's livelihood*

f) "Jemandem einen Bären aufbinden"
(lit) *to tie a bear on someone*
(non-lit.)*to put one over someone*

The idiomatic expression shown in e) has its origins in either changing the stream of water to sabotage a water mill or, more likely, removing the water ditch of a castle during a siege. Both of the possible literal meanings are nowadays infrequently used, while the metaphorical expression has become highly frequent. On the other hand, the expression used in f) had no literal origin; in fact, the origin of this expression is not well known[12]. A manual inspection, looking at a rich online etymological dictionary of German, the *Digitales Wörterbuch der deutschen Sprache* (Klein and Geyken, 2010), reveals the following frequency curve for both the expressions shown in Figure 2.4.

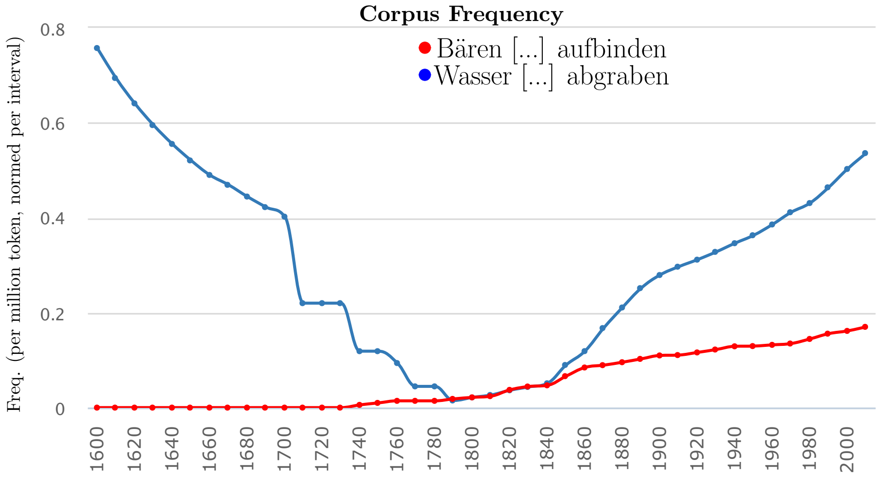


Figure 2.4: Frequency comparison for "Bär(en)[...]aufbinden" and "Wasser[...]abgraben", within a window of 10 words over the years 1600-2016, sliced and normed in 10-year intervals.

Each point in the figure is normed according to its time interval slice. The figure shows clearly that in the case of "Wasser abgraben", both of the terms co-occurred together frequently in the intervals before 1700. A manual inspection shows that the contexts refer here to the literal sense only. As time went by, the common frequency decreases reaching a low point at around 1800. Afterwards, the idiomatic usage

[12] There are multiple theories. One assumes that the idiom refers to the fact that it should be impossible to tie a bear on somebody, without that person noticing.

became more popular than its literal usage resulting in a high total frequency of both tokens. In contrast, all the corpus samples from "Bär(en) aufbinden" refer exclusively to the idiomatic expression starting at around 1800.

Since most idiomatic expressions are well known, we would like to mention the ones including PVs. A manual search, relying on *Wiktionary*[13] and *Redensarten.Net*[14] showed that PVs occur in frequent idiomatic expressions, such as:

Bären *aufbinden*, Wasser *abgraben*, Ast *ablachen* (*to laugh very much*, literal: laugh off a limb of a tree), sich einen *abbrechen* (*to try very hard*, literal: to break oneself), Den Ast *absägen* auf dem man sitzt (literal: saw off the branch we are sitting on), Mit dem falschen Fuß *aufstehen* (literal: to get up with the wrong foot, similar to: caught on the wrong foot), Hörner *aufsetzen* (literal: put horns on sb., to cuckold sb.), Radieschen von unten *ansehen* (*to die/being dead*, literal:watch the radishes from below), Jemandem etwas *abknöpfen* (*get sth. off sb.*, literal: to button off sth.), Breitseite *abfeuern* (*to be hostile/fight sb.*, literal: to fire broadside), Dampf *ablassen* (*to blow off (one's) steam*), den Löffel *abgeben* (*to die*, literal: to give away the spoon), Jemanden *abblitzen* lassen (*to snub sb.*, literal: to flash away sb.), Alte Zöpfe *abschneiden* (to do away with old shibboleths, literal: to cut off of old bunches), den Rang *ablaufen* (*to outstrip sb.*, literal: to run down the rank of sb.), Staub *aufwirbeln* (*make trouble or a sensation*, literal: to blow up dust), Rechenschaft *ablegen (to give answer to sb.*, literal: lay off accountability), andere Saiten *aufziehen* (*to change tune/to get tough*, literal: change strings.), Mit Pauken und Trompeten *durchfallen* (*to fail miserably, literal: to fail with drums and trumpets*), Wie ein Schiesshund *aufpassen* (*to keep a close watch*, literal: watch like a gun dog), Den Teufel mit dem Belzebub *austreiben* (*to replace one evil with another*), auf seinen Loorbeeren *ausruhen* (*to rest on one's laurels*), die Suppe *auslöffeln* (*to face the problem*, literal: to spoon up the soup), eine Scharte *auswetzen* (*to compensate for a loss*, literal: to patch a notch), and Daumenschrauben *anlegen* (*to put the thumbscrews on sb.*).

[13] `https://de.wiktionary.org/wiki/Verzeichnis:Deutsch/Redewendungen`

[14] `http://www.redensarten.net`

3 Methodology, Data, and Resources

3.1 Distributional Semantics

> “*You shall know a word by the company it keeps*
> J. R. Firth, *A synopsis of linguistic theory*, 1957”

The meaning of a word and its representation is crucial for this work. Hence, computational approaches require a suitable meaning representation for a word. Decades of research relied on context, as the key to approximate a word‘s meaning. The so-called Distributional Hypothesis is that “*words that occur in the same contexts tend to have similar meanings*” Harris (1954), as well as the more famous statement from Firth (1957). The meaning of a word depends on its usage, and therefore, the distribution of its contexts.

Consider the following example taken from Lazaridou et al. (2014):

a) “*We found a cute, hairy* ***wampimuk*** *sleeping behind the tree.*”

Although we have never heard of a wampimuk, based on the context of this single sentence, a reader can get a glimpse of its meaning and, with more contextual information available, an even clearer picture emerges.

Applying the Distributional Hypothesis allows us to estimate similarity between words by comparing their context distributions. Usually, this is done by working with large amounts of language data. We want to explicitly point out that similarity is not about appearing together in *the same* context. In fact, many synonyms rarely appear together in the same context, e.g., usually one uses either *football* or *soccer* but not both. On the other hand, words that appear together in many contexts are not necessarily semantically similar, e.g., collocations such as *cosmetic surgery*. Grefenstette (1994a) distinguishes first and second-order affinities, where first-order affinities describe immediate, or local, word neighbors. On the other hand, second-order affinities, describe words that share the same environments (context distribution).

In the case of *football - soccer* both words frequently share neighbors such as *game, ball, match, win*. Although they rarely occur together, both words are considered semantically similar, according to the Distributional Hypothesis, as their context distributions are similar.

The Distributional Hypothesis is the basis for statistical approaches to semantics. Methods based on this hypothesis are often called distributional methods. Furthermore, the term distributional semantics is used to refer to an entire subfield of natural language processing that applies the Distributional Hypothesis to learn and represent word meaning. Another essential point is that distributional semantic models (DSMs) have been applied successfully to a variety of different applications. Beyond word similarity, distributional semantics is used for automatic thesaurus generation, information extraction, query expansion, word sense disambiguation, sentiment, and many more; a detailed list can be found in the excellent survey from Turney and Pantel (2010).

In the last recent years, neural networks or deep learning architectures and algorithms have made impressive advances in the field of natural language processing. This development lead also to novel methods for word representation learning, as well as a growing interest for applications and evaluation techniques. Low-dimensional word representations are particularly interesting for neural architectures since they can easily be integrated as features into such machine learning systems. This has been shown to be useful, especially in the context of neural methods that rely on vector representations of words as input data. Luong et al. (2013):

> [...] *"The use of word representations [...] has become a key "secret sauce" for the success of many NLP systems in recent years, across tasks including named entity recognition, part-of-speech tagging, parsing, and semantic role labeling."* [...]

We will now describe two different techniques used to obtain word representations. Some of the following figures, used to describe the methods, have been already published in Köper (2014). More information on the subject of vector-space models can be found in the extensive literature (Turney and Pantel, 2010; Clark, 2015; Jurafsky and Martin, 2017, Chapter 15 and 16).

3.1.1 Count Vector-Space Models

In practice, distributional semantics makes use of a semantic space or a vector-space model (VSM). In such a space, points are close to each other when they are semantically similar, whereas unrelated words are more distant. The traditional way of creating such a vector-space model is done by counting contextual information. Following the terminology of Baroni et al. (2014b), we call models of this type count models as opposed to predict models (Section 3.1.2).

Vector-space models were first applied to represent a document in a high-dimensional space as part of an information retrieval system (Salton, 1971). This concept was then extended successfully to semantic tasks, representing words and pairs of words in a high-dimensional space (Deerwester et al., 1990; Rapp, 2003; Turney, 2006; Bullinaria and Levy, 2007). Context-counting models collect distributional information in high-dimensional vectors by counting context words (word co-occurrences) typically from a large corpus. The dimensions correspond to the words or explicit features in the vocabulary. This counting procedure shall be illustrated by looking at an example sentence (Figure 3.1).

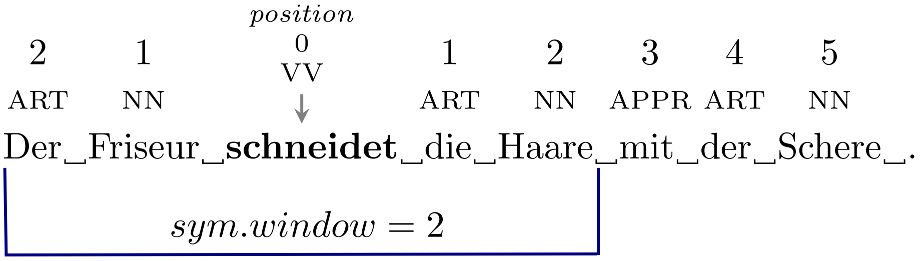
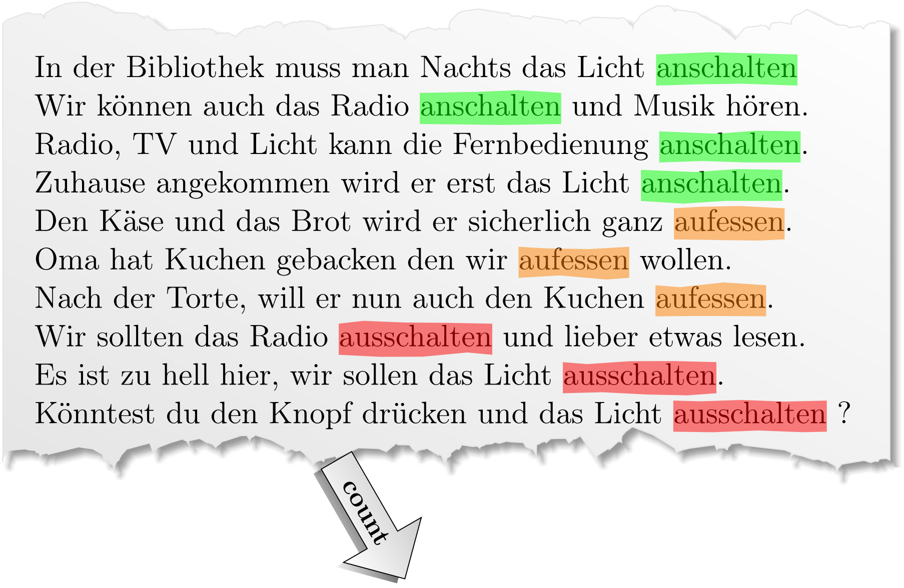


Figure 3.1: Example sentence, illustrating the usage of a window. Additionally the distance to the target *schneiden* and part-of-speech information is shown.

Let us assume we are interested in learning a distributional representation for a certain target word, in this example *schneidet (sb. cuts)*. Counting requires keeping track of every possible context word next to our target word. Given the large collection of texts, we would search for every occurrence of *schneidet* and count the co-occurrence of each of the context words (*der*, *Friseur*, *die*, *Haare*, *mit*, *der*, *Schere*).

The context could be the entire document, a paragraph, or the sentence. The majority of approaches use a smaller context, often, just a few surrounding words defined by a window. By applying a window, one can restrict the context words to the immediately adjacent neighbors. Figure 3.1 shows a symmetrical window of size 2, that is, ±2 words to the left and right of the target word. The size or type of the window depends on the goal of the representation, shorter windows capture more syntactic information, whereas longer windows tend to capture more semantic information (Jurafsky and Martin, 2017). In addition, common preprocessing steps include lemmatization, lowercasing as well as filtering of infrequent words and punctuation. Wiemer-Hastings and Zipitria (2001) enhanced the contextual information based on the word class by combining words with their part-of-speech tag. Another approach is to consider context words syntactically related to the target word (Grefenstette, 1994b; Lin, 1998; Padó and Lapata, 2007).



In der Bibliothek muss man Nachts das Licht anschalten
Wir können auch das Radio anschalten und Musik hören.
Radio, TV und Licht kann die Fernbedienung anschalten.
Zuhause angekommen wird er erst das Licht anschalten.
Den Käse und das Brot wird er sicherlich ganz aufessen.
Oma hat Kuchen gebacken den wir aufessen wollen.
Nach der Torte, will er nun auch den Kuchen aufessen.
Wir sollten das Radio ausschalten und lieber etwas lesen.
Es ist zu hell hier, wir sollen das Licht ausschalten.
Könntest du den Knopf drücken und das Licht ausschalten ?

$$
\begin{array}{l c}
 & \begin{array}{ccccccc} \textbf{das} & \textbf{Licht} & \textbf{Radio} & \textbf{und} & \textbf{Kuchen} & \textbf{Brot} & \textbf{...} \end{array} \\
\begin{array}{l} \textbf{anschalten} \\ \textbf{aufessen} \\ \textbf{ausschalten} \end{array} & \begin{pmatrix} 3 & 3 & 2 & 2 & 0 & 0 & ... \\ 1 & 0 & 0 & 1 & 2 & 1 & ... \\ 3 & 2 & 1 & 1 & 0 & 0 & ... \end{pmatrix}
\end{array}
$$

Figure 3.2: Toy example of a tiny text corpus and counts for three target words (colored) together with the count values for some of their dimensions.

Finally, the co-occurrence information for a target word is stored in a vector representation. Such a vector is usually very highly dimensional and sparse (many zero entries), since the number of dimensions equals the number of possible contextual features (often the size of the vocabulary $|V|$). Mathematically, a vector with n dimensions corresponds to an ordered list of n real numbers $\vec{x} = [x_1, x_2, x_3, ..., x_n]$. To illustrate this, we provide a large example in Figure 3.2. In this example we would count every word within a sentence of the target word. The three target words (*anschalten*, *aufessen*, *ausschalten*) are highlighted in color. The resulting vector contains one dimension for every possible context word; thus, there are 91 dimensions in this example. The resulting rectangular array structure, used to store the counts, is a matrix. The resulting counts for six of the 91 dimensions are shown in the matrix below the text. Hence, the dimensions of the count matrix are 3 × 91 with a column representing a unique context feature, whereas a row represents the actual vector of a target word. Two vector representation can only be compared with each other when their entries refer to the same contextual dimensions.

Cosine-Similarity

Vectors have many useful properties. We can easily compare two vectors by computing their distance to each other. Moreover they support mathematical operations (algebraic operations), we can add, subtract, multiply, or merge multiple vectors. The most common technique, in the context of distributional semantics, is to compute similarity between vectors by measuring their cosine similarity. Equation 3.1 shows how to compute the cosine similarity between two vectors (A and B).

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{3.1}$$

The resulting similarity measure ranges between $[-1, 1]$, with -1 meaning that the vectors are very different (opposite) and large values indicating high similarity. Two identical vectors obtain a similarity of 1. In the context of word vectors, a high frequency word would gather more corpus counts which result in a larger magnitude. The magnitude of a vector $\vec{x}$ is defined as: $\|\mathbf{x}\| := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. Unlike other distance measures (Euclidean, Manhatten distance), the cosine is not sensitive to a vectors magnitude or length since the denominator in Equation 3.1 performs a length-normalization. Hence, cosine-similarity is the normalized dot-product between two vectors. Therefore, similarity is measured by looking at the angle between two vectors.

Figure 3.3 illustrates the cosine similarity measure, based on the previous counting example (Figure 3.2). The two vectors $\overrightarrow{aufessen}$ and $\overrightarrow{ausschalten}$ point into different directions. Therefore, the red angle (cosine) between both vectors is large, result-

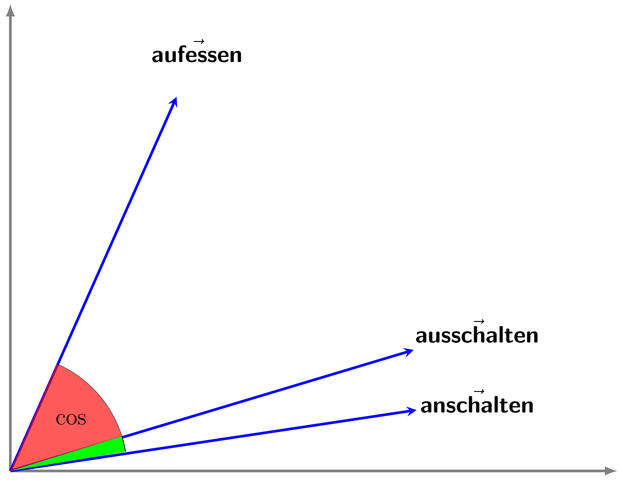


Figure 3.3: Cosine-Similarity illustration.

ing in a low cosine similarity. On the other hand the cosine between $\vec{ausschalten}$ and $\vec{anschalten}$ (green angle) is comparably smaller, since both vectors point into a similar direction. In this example, we would see that $\vec{ausschalten}$ is more similar to $\vec{anschalten}$ than to $\vec{aufessen}$, according to the cosine similarity.

Weighting

In practice, raw co-occurrence counts do not work that well. Count models often achieve much higher performance when the features are weighted according to some weighting scheme. The underlying motivation of weighting is to give more weight to infrequent or surprising events and less weight to frequent and expected events. In the context of word counting models, frequent events would refer to a context occurrence of a so-called *stop word* such as *the*, *is*, *at*, *which*, *and*. These words carry little semantic information but obtain very high frequency counts.

Point-wise Mutual Information (PMI, Equation 3.2) is a measure of association between two values of two random variables. PMI was introduced to NLP by Church and Hanks (1989).

$$\text{pmi}(x,y) \equiv \log \frac{p(x,y)}{p(x)p(y)} = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)}. \tag{3.2}$$

In the last decades, it has been applied to a variety of natural language processing tasks and has been shown to work well. The **Positive PMI** (PPMI, Equation 3.3) is a modified version of PMI, in which all mutual information values that are less than zero are replaced with zero. Bullinaria and Levy (2007) reported that PPMI performs better than a variety of other weighting approaches when measuring semantic similarity with word–context matrices.

$$\mathrm{ppmi}(x, y) \equiv \begin{cases} \mathrm{pmi}(x, y), & \mathit{if}\ \mathrm{pmi}(x, y) \geq 0 \\ 0, & \text{else} \end{cases} \tag{3.3}$$

Local Mutual Information: (LMI, Equation 3.4) (Evert, 2005) is very similar to PMI. Pointwise Mutual Information suffers from being biased toward infrequent events. Thus, PMI gives too much weight to low frequency pairs. The LMI measure tries to balance this behavior by multiplying the PMI score with the pair frequency.

$$\mathrm{plmi}(x, y) \equiv \mathrm{ppmi}(x, y) \cdot \mathrm{count}(x, y) \tag{3.4}$$

This section contains only the weighting schemes used in this thesis. Of course, many more weighting schemes are possible. Other commonly used weighting techniques include log and entropy as used by Landauer and Dumais (1997); Turney (2006) or combining term frequency with inverse document frequency (idf) (Jones, 1972). Furthermore, it is possible to normalize the matrix row- or column-wise and turn the entries into probability distributions.

Dimensionality Reduction

As mentioned already, a word-context count matrix can easily reach a very large dimensionality. Large dimensions are computationally inefficient since they require more memory and make computations, such as performing cosine-similarity, slow. Therefore, dimensionality reduction represents another popular type of transformation typically applied to count matrices. Dimensionality reduction techniques reduce the context dimensionality and approximate the original vector-space in a smaller space. The most commonly used dimensionality reduction in NLP is singular value decomposition (SVD). The idea of applying SVD to the text domain goes back to Deerwester et al. (1990). The key idea of this technique is that given a term-document or term-context matrix, similar terms should have similar vectors (rows). SVD is used to reduce the number of rows (documents or contextual features) while preserving the similarity structure among the columns.

$$
\overset{\substack{\text{Matrix X}\\ M \times N}}{\begin{array}{r|cccc|} & \textit{das} & \textit{Licht} & \textit{Radio} & \dots \\ \hline \textit{ausschalten} & 3 & 3 & 2 & \dots \\ \textit{aufessen} & 1 & 0 & 0 & \dots \\ \textit{ausschalten} & 3 & 2 & 1 & \dots \end{array}} = X = U \Sigma V^T = \overset{\substack{\text{left-singular vectors}\\ \text{U} = XX^T\\ M \times M}}{\left(\begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \begin{bmatrix} \\ \mathbf{u}_2 \\ \\ \end{bmatrix} \dots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \right)} \cdot \overset{\substack{\Sigma\\ \text{singular values of X}\\ M \times N}}{\begin{pmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{pmatrix}} \cdot \overset{\substack{\text{right-singular vectors}\\ \text{V} = X^T X\\ N \times N}}{\begin{pmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{pmatrix}}
$$

Figure 3.4: Singular Value Decomposition of a count Matrix (X) into $U \Sigma V^T$

Mathematically, SVD decomposes a matrix X into a product of three matrices $X = U \Sigma V^T$. The matrix of Σ contains only values on the main diagonal. These values are called the *singular values*. Assuming the matrix Σ is of rank $= r$, then, dimension reduction is done by simply picking the top k (with $k < r$) singular values of Σ and the corresponding columns from U and V. The matrix $U_k \Sigma_k V_k^T$ is the best approximation of the original matrix X with rank k. The resulting vector representations are usually dense, that is most entries are non-zero (as opposed to sparse representations). However, the context-dimensions are no longer explicit; thus, they are not easily interpretable. A number of studies also found that dimensionality reduction techniques, such as Principal Components Analysis or SVD, can improve the performance of vector-space models (Landauer and Dumais, 1997; Rapp, 2003; Turney, 2006; Baroni et al., 2014b).

3.1.2 Predict Vector-Space Models

More recently, predict vector representations (often just *embeddings*) have gained a lot of attention. This type of word representation was inspired by neural network language models, as introduced by Bengio et al. (2003). These language models make use of low[1] dimensional dense word representations to predict the next word in a sequence of words. The word representations were just a by-product of neural language models, which were primarily designed to assign probabilities score to sequences of words. The fact that such representations are preserved in semantically meaningful neighborhoods was noticed in Blitzer et al. (2004) and also more recently by Mikolov et al. (2013d). Furthermore, it was shown that these representations as an extra word feature can be useful for other downstream applications, such as part-of-speech tagging, named entity recognition, and chunking (Collobert and Weston,

[1] low/high dimensional is in this context a very vague term. Neural language models represent words usually with less than 100 dimensions.

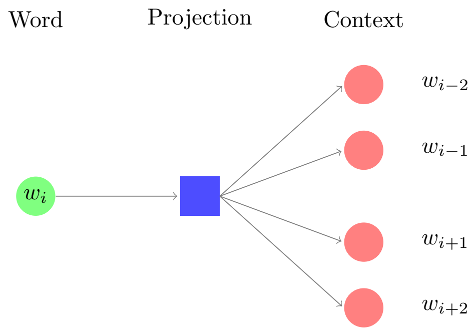


Figure 3.5: The Skip-Gram Architecture

2008; Turian et al., 2010).

Still, neural language models were computationally inefficient as the use of hidden layers together with large corpora and huge vocabularies requires many updates which is not feasible. Thus, these models were not applicable for large scale word representation learning. Later Mikolov et al. (2013a,c) introduced a simplified log-linear model with much lower computational complexity that could easily be applied to huge corpora. Sometimes, the literature refers to the models and architecture proposed by Mikolov as *word2vec*, which is the name of the tool that provides the various implementations of the models and architectures.

Skip-Gram with Negative Sampling (SGNS)

The most common predict method is the skip gram architecture together with negative sampling (SGNS), introduced by Mikolov et al. (2013c). In the same way as count models, predict models look at the contexts of target words within a defined window. Figure 3.5 illustrates this, assuming the target word w_i is at position i in the sentence. First, the models looks for the corresponding vector representation of w_i (projections). The model wants to predict the representation of each surrounding context word within a symmetrical window of size 2 using the representation of w_i. The Figure might look like the model is performing just one prediction but in fact it performs one prediction for each of the four context words.

That aside, the word representations in predict models are not updated by simply increasing context counts. In predict models, the representation of a word is changed during training based on the models' ability to predict a context word's representation given the target word representation. Since similar words occur in similar contexts, a predict model naturally learns to assign similar vectors to similar words.

Note that each word in our vocabulary has two different vector representations, a context and a target representation. This separation is technically not necessary; however, Goldberg and Levy (2014) give an intuitive explanation on its usefulness:

> [...] *Consider the case where both the word dog and the context dog share the same vector $\vec{v}$. Words hardly appear in the contexts of themselves, and so the model should assign a low probability to $p(dog|dog)$, which entails assigning a low value to $\vec{v}^\top \vec{v}$ which is impossible* [...]

More theoretically, the objective of SGNS is to maximize the conditional probability: $p(c|w)$ by setting all the weights θ (context and input representations) as to maximize the probability:

$$\arg\max_{\theta} \prod_{(w,c) \in D} p(c|w;\theta) \tag{3.5}$$

D describes the set of all word and context pairs. The conditional probability can be modeled using the so called softmax function:

$$p(c|w;\theta) = \frac{e^{\vec{v_c}^\top \vec{v_w}}}{\underset{\substack{\uparrow \\ \text{Normalize over} \\ \text{all possible contexts } (|C| = |V|)}}{\sum_{x \in C}^{|C|}} e^{\vec{v_x}^\top \vec{v_w}}} \tag{3.6}$$

Here $\vec{v}_c$ are context vectors and $\vec{v}_w$ are input representations. Equation 3.6 is very impracticable due to the normalization in the denominator which requires the iteration of all possible contexts (all $v_x \in |C|$) and the computation of $e^{\vec{v_x}^\top \vec{v_w}}$. Instead of computing the (full) softmax, Mikolov et al. (2013c) introduced an approximation technique called *negative sampling*. Here, the softmax is replaced by a sigmoid function (see Goldberg and Hirst (2017) for details); thus, the conditional probability becomes:

$$\arg\max \sum_{(w,c) \in D} \log \frac{1}{1 + e^{-\vec{v_c}^\top \vec{v_w}}} \tag{3.7}$$

Equation 3.7 makes use of target-context pairs solely observed in the data (positive examples). Unfortunately, this equation has a trivial solution, namely, assigning the same vector representation to every word and context representation.

Based on this observation, SGNS makes use of negative (random) context-target pairs. The idea of negative sampling is that we simply include additional pairs (w', c') where the conditional probability should be low. These pairs are randomly drawn from a set D_{neg} (sometimes *noise distribution*). By sampling these pairs randomly, we can assume that these pairs are unlikely to co-occur within the same context window in a sentence.

For every real observed word-context pair, SGNS creates multiple random negative samples. Equation 3.8 shows the overall training objective using negative sampling. The green part illustrates the positive samples, while red shows the negative samples. SGNS updates the weights to increase the prediction probability for positive samples. On the other hand negative pairs are updated to have a low prediction probability. The actual update is done by using a technique called stochastic gradient descent (SGD). A detailed descriptions of these neural techniques in the context of language processing can be found in Goldberg and Hirst (2017).

$$\begin{aligned}&\arg\max_{\theta} \overset{\substack{\text{Positive sample}\\ \downarrow}}{p(D=1|w,c;\theta)} \overset{\substack{\text{Negative sample}\\ \downarrow}}{\prod_{(w',c')\in D_{neg}} p(D=0|w',c';\theta)}\\ &= \arg\max_{\theta} \log\frac{1}{1+e^{-\vec{v_c}\cdot\vec{v_w}}} + \sum_{(w',c')\in D_{neg}} \log\frac{1}{1+e^{\vec{v'_c}\cdot\vec{v'_w}}}\end{aligned} \tag{3.8}$$

For comparison, the resulting equation looks very similar to Equation (4) from Mikolov et al. (2013c). In their equation $v_{w_i}v_{w_O}$ (input and output) correspond to $\vec{v_c}^{\top}\vec{v_w}$.

$$\overset{\substack{\text{positive sample}\\ \downarrow}}{\log\sigma(v'_{\vec{WO}}\cdot v_{\vec{WI}})} + \underset{\text{choose k samples}}{\sum_{i=1}^{k}\mathbb{E}_{wi}\sim P_n(w)} \underset{\substack{\uparrow\\ \text{negative sample}}}{\left[\log\sigma(-\vec{v}'_{wi}\cdot v_{\vec{WI}})\right]} \tag{3.9}$$

Hyperparameters and Randomness

SGNS comes along with many possible hyperparameters. While some of these parameters are seemingly similar to the parameters of count models, we want to highlight that their effect is not always obvious. In addition, many parameters add randomness that make reproducibility, stability, and transparency of the representation learning difficult.

Analogously to count models, when using SGNS one can define a window size, and similar to dimensionality reduction techniques one can decide on a dimensionality value. Here, it should be mentioned that in most implementations (e.g., word2vec) the window is in fact a *dynamical window*, where the window parameter defines only a maximal window. For each observation, a random window between 1 and the window parameter is chosen. Dynamical windows are designed to give more weight to adjacent neighbors as these neighbors are more likely to co-occur within the dynamical window. On the other hand, this behavior introduces randomness and makes the model less transparent.

Another technique used by SGNS is *sub-sampling*. Sub-sampling randomly removes high frequency words (stop words); this is done before computing the actual window for a given target word.

Furthermore, the number of negative samples has to be defined; this number is usually between 5 and 15. It is worth noting that frequent words are more likely to be chosen as random negative samples. The probability that a certain word w is chosen is about $P_n(w) = \frac{\text{Freq}_w}{\text{Number of Tokens}}$. The exact calculation behind the probability $P_n(w)$ is actually more complicated and involves smoothing of the frequency distribution. A detailed discussion on this can be found in Levy et al. (2015).

Another non-transparent parameter is the *learning rate*. This parameter affects how strong a learning update to the representation is. Interestingly, the learning rate is also a dynamic number. Thus, the learning rate decreases with progress. To sum up, stronger updates are carried out in the beginning of the learning (first part of the corpus), while later stages see only smaller updates to the representation. To counter this effect, one can either shuffle the corpus sentences or train over multiple corpus iterations.

Further factors, including model parameters, that affect embedding stability, as measured in terms of nearest neighbor overlap, have been studied in Wendlandt et al. (2018). Their findings show that word embeddings are surprisingly variable.

3.1.3 Relationship between Count and Predict Models

Numerous studies compared the two different types of representation learning techniques systematically (Baroni et al., 2014b; Köper et al., 2015; Levy et al., 2015; Gladkova et al., 2016). Their findings can be summarized as *"Don't count, predict!"*, as the title from Baroni and colleagues suggests. Predict models consistently showed superior performance on a variety of tasks over count models.

On the other hand, Levy et al. (2015) argued that much of the performance gains of predict models are due to certain system design choices and hyperparameter optimizations (see Section 3.1.2). Furthermore, they showed that count models can benefit from the same modifications. Similarly, Levy and Goldberg (2014b) demonstrate that SGNS implicitly factorizes a word-context PMI matrix. In a similar vein, *Glove* (Pennington et al., 2014), another popular predict model architecture, is clearly factorizing a co-occurrence matrix. Thus, the difference between count and predict models might be rather small.

In summary, it should be mentioned that count models afford more flexibility and transparency. Count vector spaces are created in multiple vector-transformation steps; these steps can create huge temporary files and it can take a lot of time. That aside, count models allow many design choices and can easily be modified. On the other hand, creating word representations with predict models is fast, memory efficient, and convenient. The dense representations are learned within a single step and perform often superior to count representations. However, predict models are clearly less transparent and the method often involves a certain degree of randomness.

3.2 Machine Learning

> “ *The question of whether Machines Can Think [...] is about as relevant as the question of whether Submarines Can Swim.*
> Edsger W. Dijkstra, *The threats to computing science*, 1984 ”

Machine learning is a subfield of artificial intelligence, where algorithms are used to learn information directly from data. The term *machine learning* goes back to Samuel (1959), who defines machine learning as:

> *"a field of study that gives computers the ability to learn without being explicitly programmed.*

Hence, machine learning models learn from the given input data. Machine learning tries to understand the structure of data and fit that data into models that can be used to make predictions over unseen events. The most common application of machine learning is to make predictions. Mathematically, this can be formulated as the need to learn a function f, that takes some input X and predicts an output Y; thus, $f(X) = Y$. The algorithms typically require a numerical representation of the input data. A single object is usually represented with multiple properties, these properties or features are stored in a vector (feature vector). Therefore, multiple objects form a matrix structure that is often called feature space. The data representation is crucial for many machine learning application; hence, the process of finding optimal feature representations is very important. Finding the best object representation for a given task is called feature engineering. The output Y can take different forms; it can be a single numerical value or a class label out of potentially many (or unknown) possible labels.

While machine learning can appear in many guises, there are two distinct major paradigms: supervised and unsupervised learning.[2] In *supervised learning*, a model learns based on example inputs together with their desired output labels. In this case, learning is done by optimizing some kind of objective function, e.g., minimize the measurement of error. Given an input, the algorithm compares its predicted output with the actual desired outputs to find errors, and modifies the model accordingly. For example, consider the task to learn a classifier that classifies emails into one of the following two categories: spam or not-spam. In a supervised setting, one gets example inputs (training data) for the two different types of emails.

Complementary to this, in *unsupervised learning*, learning is done without information about the desired output labels. Unsupervised learning is challenging, since the algorithm has to discover the important patterns or hidden structure in the data without any real output observations. Nevertheless, having annotated training data often requires human annotation which is not always feasible. Common techniques for unsupervised learning involve clustering, association learning, dimensionality reduction, and outlier detection. As a result of learning based on given input data, machine learning techniques can suffer from *overfitting*. Overfitting occurs when a model learns from the irrelevant information (noise) in the data set. Here, the model would perform very well on the training data but will not generalize well on unseen data.

[2] These two paradigms represent the common broad distinction. A more fine-grained distinction includes semi-superives, active and reinforcement learning.

3.2.1 Algorithms

We will now briefly describe the machine learning algorithms, that are used in the experimental sections of this thesis. While many algorithms are applicable to multiple tasks (e.g., classification and regression) we divided them according to the applied use cases in this thesis. In addition, this section provides only a short and rough description of each algorithm, a more detailed explanation on the algorithms can be found in the extensive literature (Duda and Hart, 1973; Mitchell, 1997; Bishop, 2006).

Classification

Classification is a supervised machine learning technique where the goal is to predict a category or class from some given inputs.

Decision Trees represent graph or tree structures used for classification. Learning is done by recursively splitting the datasets according to the features most useful for predicting the desired output. The core algorithm for decision trees from Quinlan (1986) relies on entropy and information gain to decide which of the features are the most relevant. This splitting procedure builds a tree, in a top-down manner. In this tree the leaf symbols represent the possible categorical class values and each branch represents the outcome of a feature question. A classification result is derived in a similar top-down manner, by passing the data to the first node in the tree (root), and testing the branch conditions. Figure 3.6 shows a tiny example tree for spam detections. It can be seen that decision trees are very transparent and implicitly perform feature selection. Decision trees are commonly used in ensemble methods

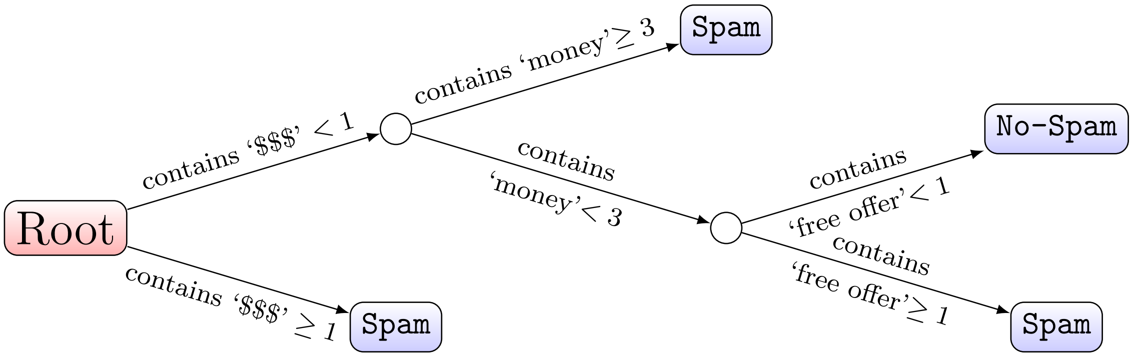


Figure 3.6: Small Example Decision Tree for Spam detection.

called **Random Forests**. Here, multiple trees are combined. Unlike default decision trees, an individual tree is built by splitting among a random subset of the entire features. In this way, up to a few hundred sub-trees are learned. After learning, the random forest can be applied to new examples by predicting the outcome of each individual sub-tree and predict the overall majority. While being less transparent, random forests are usually more accurate than single decision trees since they lower the risk of overfitting and generalize better on unseen data.

In addition to classification, decision trees and random forest can potentially be used for regression in the same way. Here, a leaf nodes represent continuous values.

Support-Vector-Machines: another popular instance of supervised machine learning methods are support vector machines (SVMs). While theoretically also applicable to regression tasks, SVMs are used frequently for classification tasks. SVMs aim to separate two classes by learning a hyper-plane that differentiates both classes. Intuitively, one can think of the hyper-plane as a high-dimensional line between both classes. While there are potentially unlimited possible hyper-planes, SVMs aim to learn the optimal one. The optimal hyper-plane is the one that segregates the two classes best and provides the largest margin as possible from the data points (support vectors) of either class. The hyper-plane is also used as a decision boundary. To illustrate, Figure 3.7 applies SVMs for a binary classification between fruit and

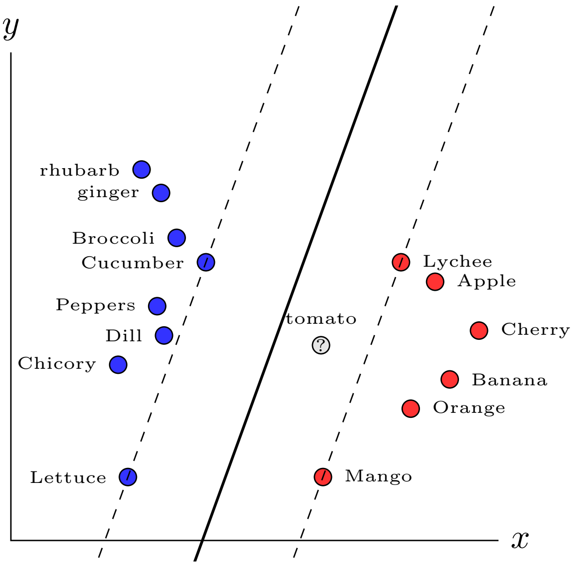


Figure 3.7: Example Support-Vector-Machines for Binary Classification Fruit vs Vegetables.

vegetables. Each point represents a support vector, except for tomato (unknown test instance). The hyper-plane with the largest margin to both of the groups is the thick black line. The method so far is only applicable to binary (two-class) scenarios. For that reason, there are multiple ways to transform a binary SVM into a multiclass SVM (Duan and Keerthi, 2005). A common approach to transform the multiclass setup into a binary classification is done by transforming the setup into multiple one-versus-all settings or into multiple one-versus-one settings. Beyond linear classification, SVMs can be extended by applying a non-linear mapping (kernel trick) of the input data into another high-dimensional feature spaces. Figure 3.8 illustrates this with an example, where the one-dimensional space is not linearly separable but the higher dimensional space is.

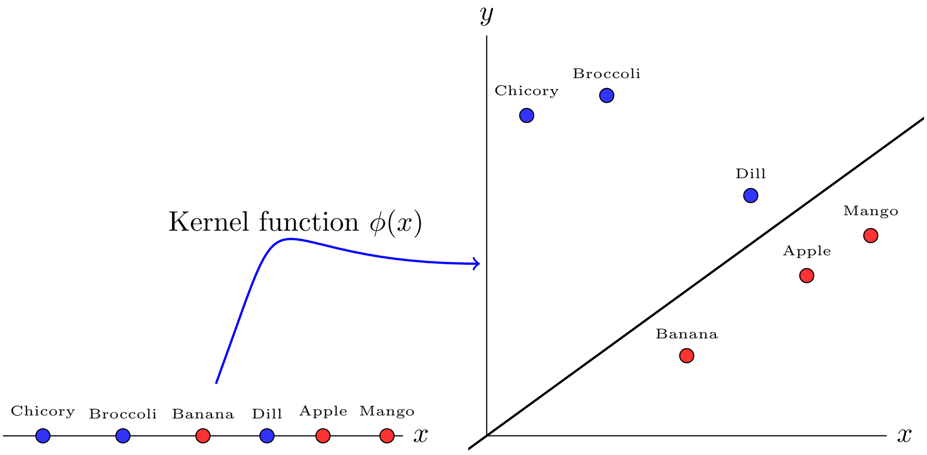


Figure 3.8: Illustration of the Kernel Trick: Projection to a Higher Dimensional Space.

Multinomial Naive Bayes: A common supervised technique, especially for text classification, is the Multinomial Naive Bayes (MNB) (McCallum and Nigam, 1998). It is a supervised probabilistic learning method. It represents a text or a document using a *bag of words* representation. In this representation, the occurrence of each word is stored while the order of words is considered irrelevant. In training, we are given a fixed set of classes $C = \{c_1, c_2, ..., c_n\}$ and a collection of documents together with their desired class labels. We assume that we are given an unknown document d that needs to be classified. During training, we compute the total probabilities for each class c_x; thus, $P(c_x)$ measures how often class c_x occurs in total. In the same way we compute the probability between a document d and class c_x by calculating $P(t_i|c_x)$ for every term (t_i) that occurs in document d_k. The probabilities are multiplied together, as shown in Equation 3.10, and the final class prediction is the c_x

with the highest probabilities.

$$P(c_x \mid d) \propto P(c_x) \prod_{i=1}^{n} P(t_i \mid c_x) \tag{3.10}$$

A single missing or non-observed word would lead to a zero probability and therefore render $P(c_x \mid d)$ in Equation 3.10 to zero. Because of this reason smoothing, such as add-one-smoothing (or Laplace smoothing) is applied; here, each conditional word probability is modified by adding $+1$ to its frequency count in order to avoid zero probabilities. Nevertheless, the multiplication of many probabilities in Equation 3.10 can lead to very small numbers (floating-point underflow). For this reason it is better to apply the sum of the logs instead of the multiplication; the final equation becomes Equation 3.11:

$$P(c_x \mid d) = \log P(c_x) + \sum_{i=1}^{n} x_i \cdot \log P(t_i \mid c_x) \tag{3.11}$$

As can be seen in this equation, the features (words) contribute independently to the overall decision. That is regardless of any possible interaction between the features.

Regression

A regression task is when the machine learning model needs to output a numerical value. In its linear form, a model tries to fit a linear equation to the observed data. This is often done by using the least-squares approach. Here, the model tries to minimize the sum of the residuals, a residual is the difference between observed value and the estimated value. Thus, the model minimizes $S = \sum_{i=1}^{n} {r_i}^2$, with r_i being the ith residual. Linear regression has the form $Y = a + bX$, with Y being the output or dependent variable and X being the independent or explanatory variable. Figure 3.9 shows such a linear regression model between two variables. In most real problems, we are facing multiple linear regression, which is based on more than just one explanatory variable. Moreover, regression analysis can be applied in both directions, it can be used to make predictions over unseen data, as well as to quantify the strength of the relationship between features and output.

Feed-Forward Neural Network: In this thesis, we sometimes rely on a feed-forward neural network for regression. These models were inspired by the way biological nervous systems (the brain) process information. The human brain consists of neurons that send activation signals to each other. The elements of the net-

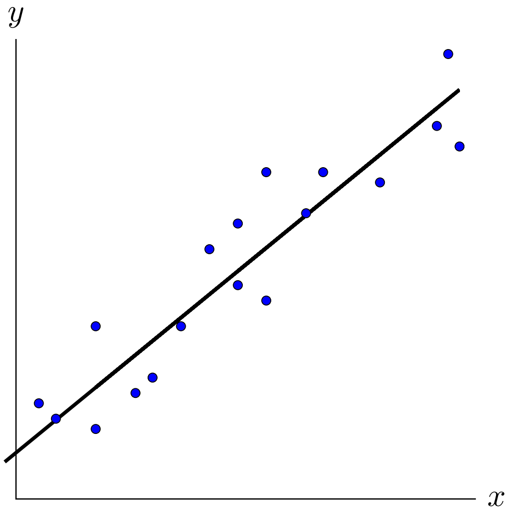


Figure 3.9: Linear Regression Example - fitting the line to the data.

work are also called neurons and the term neural network refers to the connection or communication between the neurons. The whole network is usually organized in layers. The architecture can also be seen as a graph, where the nodes correspond to neurons. Each layer consists of one or more nodes. The edges between the nodes indicate the flow of information from one node to the next. In this case, a directed acyclic graph is called *feedforward*, while networks with cycles are called *recurrent*. In all the feed-forward networks, information flows only from the input to the output, which is left-to-right in Figure 3.10. Figure 3.10 shows a multi-layer feed-forward neural network architecture with four layers including an input layer, two hidden layers and one output layer. As can be seen in the figure, each neuron in one layer has directed connections to the neurons of the subsequent layer. Each neuron receives one or more inputs (carried on the incoming-edges). Usually, the inputs of each node are weighted by weights w_{ij} and are shifted by a bias factor specific to each neuron. Then, the sum of these weighted inputs (together with the optional bias) is passed through a non-linear function (φ) known as an *activation function*. This function introduces non-linearity into the network; hence, networks have the ability to learn non-linear relationships. While there are many choices for an activation function, we apply rectified linear units in our experiments, see (Goldberg and Hirst, 2017) for more detailed information. The resulting neuron output (on its outgoing-edges) is then either used as input for the succeeding neuron or is used to compute the entire network output (Bebis and Georgiopoulos, 1994). In the case of regression, the final output represents a single continuous number.

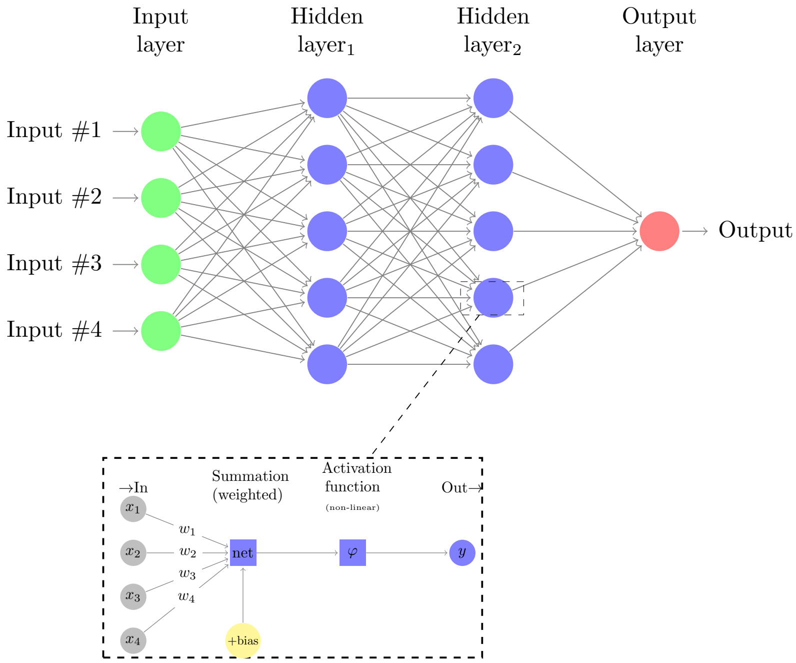


Figure 3.10: Feed Forward Neural Network with two Hidden Layers shown in the top. Bottom part zooms into a single Neuron.

Clustering

Considerable work in unsupervised machine learning has focused on the task of clustering. Here, a set of data points is grouped into a usually fixed number of groups. An important distinction can be made between hard and soft clustering. Hard clustering assigns each data point to exactly one cluster. This is in contrast to soft clustering, which results in a fuzzy (soft) membership distribution between a data point and all clusters.

One of the simplest algorithms is **k-means** (MacQueen, 1967) clustering. K-means performs a hard clustering on a set of data points $x_i = (x_1, x_2 ... x_n)$, each being a d-dimensional vector, and a cluster granularity k that defines the number of clusters. In the very beginning, k-means is initialized by guessing k random positions in the high-dimensional space. Next, each data points is associated with its nearest cluster point according to some distance measure (e.g., Euclidean or cosine). Afterwards, the position of the cluster will be updated according to its new members. This is done by calculating the mean of the assigned data points. This mean is called the *centroid*. Equation 3.12 shows how to compute the centroid μ at time $t + 1$ based on

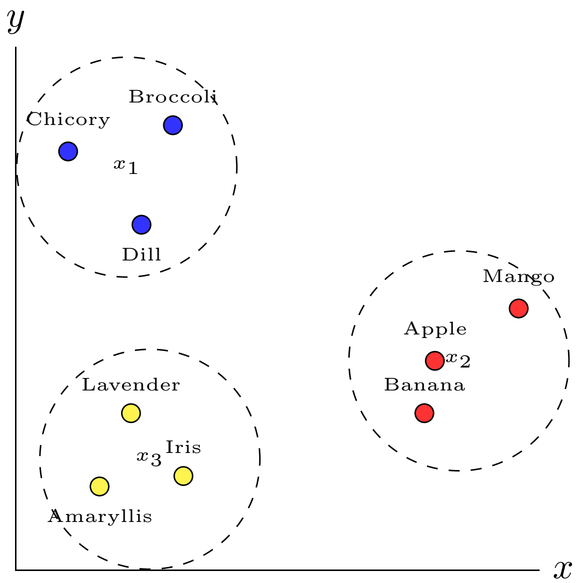


Figure 3.11: Clustering Flowers, Vegetables and Fruits by applying K-Means with $k = 3$.

the data points in G at time t.

$$\mu_i^{(t+1)} = \frac{1}{|G_i^{(t)}|} \sum_{x_j \in G_i^{(t)}} x_j \tag{3.12}$$

The element-centroid assignment together with the centroid update is iteratively repeated until a certain convergence criteria is fulfilled, e.g., the centroid does not change its position anymore. Figure 3.11 illustrates an example of k-means, where nine elements are grouped into three clusters, the final centroid position is marked with an x_k.

An essential point and a potential weakness of k-means is the very first step, namely choosing the initial cluster positions in the initialization (the seeding). This step can affect the clustering quality; the literature, therefore, contains more sophisticated methods to locate the initial cluster centers, e.g., *K-means++* Arthur and Vassilvitskii (2007), *Random Partition* or *Forgy* (Hamerly and Elkan, 2002).

Furthermore, there are various extensions of k-means. **Fuzzy C-Means** (FCM) (C. Dunn, 1973; Bezdek, 1981) extends k-means by allowing fuzzy memberships, thus performing a soft clustering. In c-means, the centroid of a cluster is still the mean of all the elements but weighted by the degree of belonging to the cluster. The output of c-means is a membership matrix where an entry e_{ij} corresponds to the membership degree of element e_i belonging to cluster j. This membership degree is a real value between zero and one; all membership degrees for a given element e_i sum up to one. Another, less known, extension is **X-Means** (Pelleg and Moore, 1999). This extension can be seen as a non-parametric[3] version of k-means where the cluster granularity k is found automatically by the algorithm. The algorithm searches over many values of k and scores each clustering model using the so-called Bayesian Information Criterion (BIC) (Schwarz, 1978). Based on this value, the algorithm decides if a cluster should be divided into two clusters or not. X-means stops when it has reached the highest BIC score.

Another less popular clustering technique used in this thesis is **BIRCH** (Zhang et al., 1996), a hierarchical clustering method. BIRCH stands for balanced iterative reducing and clustering using hierarchies. It is a hierarchical clustering method with the ability to incrementally and dynamically cluster incoming data points. While reading new data points, it builds a tree structure dynamically on top of the data points, called a CF tree, which is a height balanced tree. The final tree structure depends on a branching factor, as well as on a similarity threshold. BIRCH can be used with a predefined clustering granularity; in this case, a global clustering is carried out using the found subclusters. On the other hand, one can use the automatically found sub-clusters; in this case, BIRCH determines the cluster granularity automatically analgoues to X-Means.

A different approach to clustering is the use of topic models. Topic models aim to find the hidden topics that occur in a collection of texts or documents. The most well-known technique is **latent Dirichlet allocation** (Blei et al., 2003)) (LDA). Given a predefined number of topics, LDA assumes that each document is the result of a generative process, where words are associated to topics with a certain probability. Furthermore, a document is a mixture of different topics. Given only the set of documents, LDA tries to backtrack the generative processes and searches for topics that are likely to have generated actual observed documents. Topic models can be used for multiple purposes such as grouping documents into topics, but they can additionally be used for exploration or summarization. In contrast to LDA, the **hierarchical Dirichlet process** (HDP) (Teh et al., 2004) extends LDA by automatically inferring the number of underlying topics.

[3] Nonparametric only with respect to the cluster granularity.

3.2.2 Evaluation Measures

There are multiple ways of evaluating machine learning algorithms. This section provides explanation on the different evaluation measures used in this thesis. In short, all the measures output a numerical value with high values indicating good performance. More detailed descriptions and various examples can be found in Manning and Schütze (1999); Manning et al. (2008).

Classification

In most of the experiments, we apply classification. In classification, we are observing predictions from the model together with the desired outcomes. All the evaluation measures for classification rely on count information. Specifically, the number of true positives (tp), true negatives(tn), false positives (fp) and false negatives (fn). The four classes can be distinguished by looking at the various cases shown in Table 3.1.

		Actual class	
		p	**n**
Predicted class	**p′**	True Positive	False Positive
	n′	False Negative	True Negative

Table 3.1: Table of confusion, actual, and predicted class

Note that these four values are defined for each individual class. Table 3.2 illustrates these counts using 27 elements and three classes.

		actual						
		Flowers	Vegetables	Fruits	TP	FP	FN	TN
pred.	Flowers	5	2	0	5	2	3	17
	Vegetables	3	3	2	3	5	3	16
	Fruits	0	1	11	11	1	2	13

Table 3.2: Example of a confusion matrix with three classes.

The most intuitive measure is accuracy (Acc); given a collection $\mathcal{D}$, it measures the ratio of the number of correct classifications to the total population. Thus it sums up all the true positives (tp) across every class c:

$$\text{Accuracy} = \frac{\Sigma_{c \in C}\, tp^{(c)}}{|\mathcal{D}|} \tag{3.13}$$

Accuracy is an intuitive and interpretable measure that is often used to report results for binary classification. On the other hand, accuracy can be misleading for imbalanced datasets. If one class is highly frequent, a weak model might obtain a very high accuracy since it assigns most of the instances correctly with the most frequent label. In addition, accuracy is not very useful for cases with more than two classes, since it gives no information about predictive power of the model for the individual classes. Therefore, we usually rely on other (or additional) evaluation measures, such as Precision, Recall, and F-measure. Precision is a measure from information retrieval; it measures the fraction of relevant instances among the retrieved instances:

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3.14}$$

A high precision can be obtained when the classifier is very careful, e.g., when the classifier tries to minimize the false positive predictions. For example, when the model predicts a certain class correctly once, here, $fp = 0$ and $tp = 1$ leading to the maximum precision. Hence, precision is always used together with recall (also sensitivity) which measures the fraction of relevant instances that have been retrieved over the total amount of relevant instances. A high recall can be obtained when the classifier returns everything to minimize the number of missing results (false negatives).

$$\textbf{Recall} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false negatives}} \tag{3.15}$$

In a retrieval or recommendation setup, it is sometimes possible to use recall and precision more generously by applying **recall/precision at k** (or out-of-k). Here, k is a user defined value. This measure would look at the top-k elements and compute the proportion of recommended items in the top-k set that are relevant (precision) and the proportion of relevant items found in the top-k recommendations (recall). In this thesis, we use recall-at-k in a setup with a huge amount of elements and only one true relevant element (Section 4.3). In such a setup, it is unlikely that the system will output the correct element as first recommendation. Still, to access the systems performance it is desirable to know that the correct element is at least among the top-k predictions, which is captured by using recall-at-k.

In many cases, a good classifier should obtain both high precision and high recall. This can be measured by reporting the f-measure (F_1, f-score), which measures the harmonic mean between precision and recall.

$$\textbf{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.16}$$

When dealing with many classes, it is often convenient to compute an average across classes for each of the evaluation metrics (Precision, Recall, F-Score). This average can be computed either by creating a macro or a micro average. The **macro-average** is simply the average of all the class specific evaluation metrics. Here, each class contributes equally to the overall score, independent of the sample size of each class. **Micro-average**, on the other hand, is computed by summing up all the individual true positives, false positives, and false negatives across all classes and, then, computing the corresponding evaluation metrics. Micro-average takes the sample size into account and is biased toward the larger classes.

Soft-Clustering

The case of soft-clustering is very difficult when it comes to evaluation. In contrast to (hard) clustering or classification, a single element can occur multiple times in different groups. The (soft) clustering method has to decide on the degree of fuzziness for each element. Therefore the total number of elements can be higher or even lower than the one in a gold resource. Thus, an evaluation measure has to penalize a clustering that generates too much or one that misses elements.

In our experiments, we rely on the fuzzy extension of **B-Cubed** from Bagga and Baldwin (1998) as an evaluation measure, because it is (a) a pair-wise evaluation, which is considered as most suitable for soft clustering evaluations, and (b) distinguishes between homogeneity and completeness of a clustering and, thus, resembles an evaluation by precision and recall. Amigó et al. (2009) demonstrated the strengths of B-Cubed, and a similar version has been used in SemEval 2013 for Word Sense Induction (Jurgens and Klapaftis, 2013). Pair-wise precision P determines the homogeneity of a cluster analysis. Precision looks at all the pairs of elements that occur together in a cluster (Equation (3.17). Here, the numerator becomes zero, in cases where two elements occur together in a cluster but not in a gold class (false positive). Additionally, the fraction becomes smaller than one in cases where we falsely put elements too often together in a cluster. Pair-wise recall R determines the completeness of a cluster analysis. Here, we look at all the pairs of elements that should be grouped together according to the gold-standard class g, cf. Equation (3.18). The resulting value is zero when the clustering misses pairs (false negatives). In addition, the value is decreased when a pair should be grouped together more frequently than in the clustering. The overall B-Cubed precision and recall scores are the averages over all the element-wise scores. Both of the measures can be combined by computing the harmonic mean (f-score).

$$P(e,e') = \frac{min(|c(e) \cap c(e')|, |g(e) \cap g(e')|}{|c(e) \cap c(e')|} \tag{3.17}$$

$$R(e,e') = \frac{min(|c(e) \cap c(e')|, |g(e) \cap g(e')|}{|g(e) \cap g(e')|} \tag{3.18}$$

Correlation

Correlation evaluation is always applied when two variables are measured. Thus, we have two d-dimensional vectors with numerical values. A common use case for correlation measures in NLP is to measure a model's ability to provide a similarity score between two words. This is done by comparing the degree of similarity, against human judges. Here, the two most used measures are Pearson Correlation (r) and Spearman's rank correlation ρ (rho).

Pearson correlation is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{3.19}$$

Here we have n values across two datasets X_i, ..., X_n and Y_i,..., Y_n. $\bar{X}$ and likewise $\bar{Y}$ are defined as the mean of all values in this dataset $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Applying Pearson correlation on ranked variables results in the **Spearman's rank correlation** (Siegel and Castellan, 1988). In the case where there are no ties in the ranking, it can be computed using the following equation:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}. \tag{3.20}$$

Here, d_i is the difference between ranks $d_i = x_i - y_i$. Unlike Pearson, Spearman's correlation is only concerned with the order of elements. Both of the correlation measures can take values from +1 to -1. A result of +1 indicates a perfect or identical correlation. Zero indicates no correlation and a result of -1 indicates a perfect negative correlation.

3.3 Concreteness and Abstractness

As explained in Section 2.3.1, metaphors can be seen as a transfer from a well-understood or more familiar concrete domain into a less-understood or more abstract domain. While this conceptional mapping can be performed without effort by humans, it represents a challenge for computational models. Earlier work on automatic metaphor detection treated the problem similarly to word sense disambiguation. The first cognitive motivated modeling approach was introduced by Turney et al. (2011). Their method showed that metaphorical word usage correlates with the degree of *abstractness* of a word's context. An abstract word is defined by a word that refers to something that we cannot perceive with our senses, as opposed to concrete words. For instance, *banana*, *yellow* and *to taste* are concrete while *economics*, *differentiate* and *trustworthy* are abstract. In this representation, the abstractness of a word or concept is located on a one-dimensional scale ranging between abstract and concrete.

In psycholinguistics, abstractness is commonly used for concept classification (Barsalou and Wiemer-Hastings, 2005; Hill et al., 2014; Vigliocco et al., 2014). In computational work, abstractness has become an established feature for the task of automatic detection of metaphorical language. To illustrate this phenomena, consider the following four example sentences:

a.1) (lit.) "Den Lippenstift kannst du dir **abschminken**."
*You can **remove** the lipstick.*

a.2) (non-lit.) "Den Job kannst du dir **abschminken**."
*You can **forget about** the job.*

b.1) (lit.) "Die Schnitzel in Mehl wenden und etwas abklopfen."
*Turn the schnitzel in flour and **hammer/tap** them.*

b.2) (non-lit.) "Das Thema sollte man auf juristische Fallstricke **abklopfen**."
The topic should be ***checked*** *for legal pitfalls. (lit: the topic should be knocked...)*

For both of the verbs, the literal context contains concrete words (lipstick, schnitzel, flour) whereas the metaphorical usage appears with more abstract terms (job, topic, legal pitfalls).

The abstractness information itself is typically taken from a dictionary, created either by manual annotation or by extending manually collected ratings with the help of supervised learning techniques that rely on word representations. While potentially less reliable, automatically created norm-based abstractness ratings can easily cover huge dictionaries. Often, the abstractness information is divided according to word classes and, then, given to a supervised classifier. It has been shown that abstractness information for nouns provides the most useful information (Turney et al., 2011). We confirmed this finding for German PVs (Köper and Schulte im Walde, 2016).

Furthermore, abstractness was frequently used for the detection of English metaphors (Turney et al., 2011; Dunn, 2013; Beigman Klebanov et al., 2015; Dinh and Gurevych, 2016). The work of Tsvetkov et al. (2014) relied on bilingual dictionaries to translate between English and multiple languages to make use of their (English) abstractness ratings; their experiments include Russian, Farsi, and Spanish.

We presented multiple contributions to the task of automatic extensions of such norms across German, English, and Estonian. An overview of our approaches with a short summary is given in Table 3.3 and will be discussed in detail in the next section.

Reference:	Köper and Schulte im Walde (2016)
Train Ratings:	4: Arousal, Abstractness/Concreteness, Imageability and Valency
Train Language:	German, enriched with a small set of machine translated English ratings.
Method:	Semantic Orientation from Association (Turney and Littman, 2003)
Contribution:	First automatically extended ratings of affective norms for German. Furthermore, the ratings were a core feature for the detection of non-literal PV usage.
Created Resource:	300 000
Resource Language:	German
Reference:	Köper and Schulte im Walde (2017c)
Train Ratings:	1: Abstractness/Concreteness
Train Language:	English
Method:	Comparison of various Methods for extending ratings.
Contribution:	Our study revealed that a feed forward neural network can be used as efficient method. Exploiting recent advances in word representation learning, we additionally showed that ratings can be learned for phrases and individual word senses.
Created Resource:	3 million
Resource Language:	English
Reference:	Köper et al. (2017)
Train Ratings:	13: Anger, Arousal, Anticipation, Abstractness/Concreteness, Disgust, Dominance Fear, Happiness, Joy, Sadness, Surprise, Trust Valency
Train Language:	English
Method:	Feed forward Neural Network
Contribution:	We created domain (Twitter) specific ratings that we identified as one of our core features in a shared task on emotion intensity predictions. Here our system relying on these ratings ranked 2nd best (out of 22 systems).
Created Resource:	13 × 1.6 million=21 million
Resource Language:	English
Reference:	Aedmaa et al. (2018)
Train Ratings:	1: Abstractness/Concreteness
Train Language:	Estonian, automatically translated from English
Method:	Semantic Orientation from Association (Turney and Littman, 2003)
Contribution:	First automatically extended ratings of affective norms for Estonian. Furthermore, the ratings significantly contributed to the detection of non-literal usage of Estonian PVs.
Created Resource:	243 675
Resource Language:	Estonian
Reference:	Köper and Schulte im Walde (2018)
Train Ratings:	9: Anger, Arousal, Abstractness/Concreteness, Disgust, Fear, Happiness, Joy, Sadness, Valency
Train Language:	English
Method:	Feed forward Neural Network
Contribution:	We performed the first cross-lingual propagation relying on multi-lingual word representations.
Created Resource:	9 × 2.2 million=19.8 million
Resource Language:	German

Table 3.3: Overview of our Approaches on Automatically extended Affect and Emotion Norms.

3.4 Automatic Extension of Affective Norms

Abstractness and concreteness are often collected together with other psycholinguistic attributes. The umbrella term **affective norms**[4] is used to refer to a variety of these attributes. A large subset of concrete words have a high **imageability**; these are words that refer to things that we can actually see. **Valence** determines the pleasantness of a word (*gift* vs. *punishment*). **Arousal** describes the intensity of emotion provoked by a stimulus (*alert* vs. *calm*). Information about the affective meaning of words is used by researchers working in multiple fields, such as sentiment analysis, metaphor detection, word processing, and lexical decision.

A collection of affective norms is usually carried out by presenting a word or a phrase to human annotators. The annotators have to decide on a numerical value within a predefined scale. The final score is, then, the average score over all annotators. Currently, a number of databases offer affective norms for words in different languages, including English (Altarriba et al., 1999; Bradley and Lang, 1994; Stevenson et al., 2007; Warriner et al., 2013), German (Võ et al., 2006, 2009; Lahl et al., 2009; Kanske and Kotz, 2010; Schmidtke et al., 2014), Spanish (Redondo et al., 2007), and Finnish (Eilola and Havelka, 2010).

Besides these, there are a number of resources that focus on a single rating type only: Brysbaert et al. (2014) collected 40 thousand abstractness ratings for English. The MRC Psycholinguistic Database[5] contains roughly eight thousand abstractness ratings. In spite of this, all of these resources cover only a small proportion of a language due to the fact that human annotators are required to obtain ratings. While human-annotated data is very reliable, this procedure has some clear drawbacks. It requires many annotators which is an expensive and time consuming procedure, especially for a collection with a large vocabulary. To illustrate this with a few numbers, we want to look at the collection from Brysbaert et al. (2014). To the best of our knowledge, to date, the collection of abstractness norms from Brysbaert is the largest one carried out. Before trimming, they had roughly 63 thousand stimuli (words, two words) and aimed for an average of 25 annotators per stimuli. The final collection resulted in a total of $\approx$ 2.4 million ratings distributed over 6 thousand tasks. Each task was paid 0.75$, thus, the collection cost at least 4.500$ and took

[4] *Abstractness/Concreteness* is not related to *affect*. However, following the usage of the term *affective norms* of Lahl et al. (2009); Kanske and Kotz (2010) we refer to all types of word norms by using the term *affective norms*

[5] `http://www.psych.rl.ac.uk/`

roughly three months.

That aside, the few existing manually created collections for the German language contain less than 3 thousand words. In addition, most of the resources focus on nouns only and, therefore, lack other word classes such as verbs or adjectives. These limitations in the coverage of the existing affective norms led to the use of computational methods that automatically extended manually created lexicons by applying machine learning. Several approaches have been proposed to combine distributional word representations (see Section 3.1) with supervised machine learning methods to extend affective norms (Turney et al., 2011; Tsvetkov et al., 2014; Recchia and Louwerse, 2015; Vankrunkelsven et al., 2015; Köper and Schulte im Walde, 2017c; Sedoc et al., 2017).

We will now describe two supervised techniques used automatically to extend an affective norms resource.

3.4.1 Using Semantic Orientation from Association

The first method that we will explain was introduced by Turney and Littman (2003). The algorithm was originally designed for rating a word according to its semantic orientation (positive or negative). Later, Turney et al. (2011) applied the same algorithm to distinguish between concrete and abstract words. The algorithm uses distributional semantics (vector representations of words) and learns to assign a rating score to unseen words based on other known labeled training instances. Using this method, they were able to learn abstractness ratings for 114 501 English words.

We applied the same algorithm to German affective norms (Köper and Schulte im Walde, 2016) and to Estonian abstractness norms (Aedmaa et al., 2018). We will now describe the procedure for the German ratings in more detail. First, we collected the available resources for German. Since there was little available data, we extended the data by translating some of the English ratings into German. Finally, we merged all the resources together. In more detail, the algorithm that assigns a rating score requires labeled training data. Therefore, first, we collected several affective ratings for German. Table 3.4 lists all the available resources together with the ratings, that we used. As can be seen, not every resource contains all the four types of ratings. In addition, we decided to translate some of the English abstractness ratings from MRC and Brysbaert et al. (2014) to increase the number of the available training instances. Here, we computed the intersection of both of the resources and

Source	Words	Abs.	Ar.	Val.	Img.
Võ et al. (2009)	2902	✗	✓	✓	✓
Lahl et al. (2009)	2654	✓	✓	✓	✗
Kanske and Kotz (2010)	1000	✓	✓	✓	✗
Schmidtke et al. (2014)	1000	✗	✗	✗	✓
MRC∩Brys *(EN → GER)*	3266	✓	✗	✗	✗
# Unique words		5237	4848	4848	2901

Table 3.4: Manually created German Resources of Affective Norms used for our automatic Extension.

used a translation tool[6] to translate the words from English to German. Missing and double translations were removed from the list, resulting in a final list of 3 266 additional words together with their abstractness ratings. We finally computed a total set of words (=unique words) for each rating type by mapping all ratings to the same scale, namely $[0, 10]$. Our scale is different to the one used by Turney et al. (2011). We use low numbers to indicate abstractness (0) and high numbers to indicate concreteness (10). We did this mapping by using a continuous function (see Equation 3.21). This function maps numbers from an interval $[min, max]$ to a new interval $[a, b]$. We set $a = 0$ and $b = 10$. We computed mean values in the case of overlapping words. For every rating type we divided the number of unique words randomly into two sets: training data (90%) and test data (remaining 10%). For the abstractness set, we made sure that the test data contains only human labeled data and not the translated ratings. The test data is later used to validate the algorithm by comparing the original ratings with the ratings created by the algorithm.

$$f(x) = \frac{(b-a)(x-min)}{max-min} + a \tag{3.21}$$

Algorithm: the core idea of the algorithm from Turney and Littman (2003) is that the degree of abstractness (or arousal, valence, imageability) can be expressed by comparing a given word w_i with a list of positive and a list of negative paradigm words. Each word is represented by a high-dimensional vector (based on context counts). Then, a rating score $R(w_i)$ is computed by simply calculating the similarity with all the positive paradigm words minus the similarity with all the negative paradigm words:

[6] Translation was done by applying the following java-google-translate-text-to-speech API: `https://code.google.com/p/java-google-translate-text-to-speech/`

$$R(w_i) = \sum_{p_j \in positive} sim(w_i, p_j) - \sum_{n_j \in negative} sim(w_i, n_j) \tag{3.22}$$

Similarity is measured using the cosine distance. The algorithm begins with an empty set of paradigm words and adds one word at a time to the paradigm list, alternating between adding a word to the positive paradigm words and then adding a word to the negative paradigm words. At each step, we add the paradigm word that results in the highest Pearson correlation with the ratings of the training data. This is a form of greedy forward search without backtracking. The algorithm terminates when 20 paradigm words have been added to both of the lists (after 40 iterations). Finally, we assign each word in our vocabulary a rating score by using Equation 3.22. Then, this score is rescaled to a numerical number within $[0, 10]$ by using the function from equation 3.21.

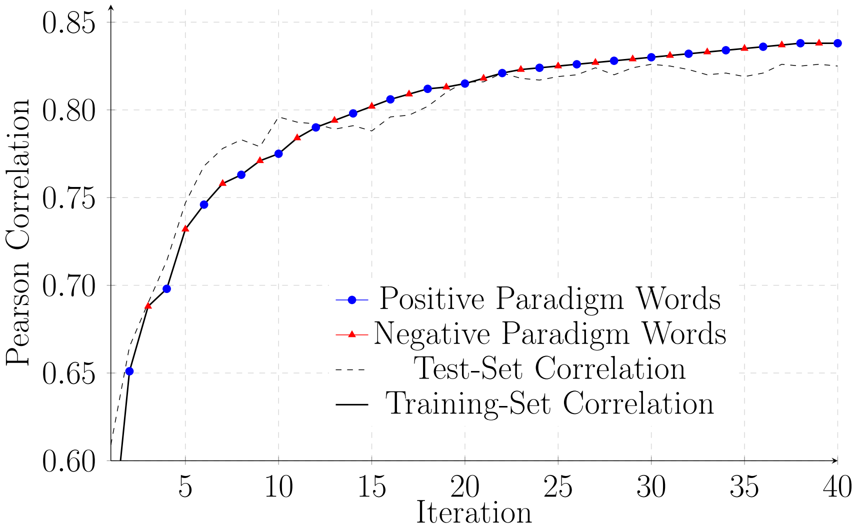


Figure 3.12: Example Progress - Train and Test Correlation for learning Abstractness.

To validate how well the algorithm works on unseen words we simultaneously measure the Pearson correlation with the (unknown) test data ratings. Note that these ratings did not influence the paradigm word selection (training process). Figure 3.12 shows the increasing Pearson correlation for the training and test data when the paradigm words are added with regard to the abstractness ratings. The figure shows that both of the correlations obtain already a high correlation (> 0.80) after only 15 iterations. While the training correlation increases with each iteration, it can be observed that in some cases the next paradigm word decreases the correlation for the test data. Moreover, looking at the added paradigm words we find a diverse set of words from different semantic topics within both of the lists; concrete paradigm words include *Schulbuch (schoolbook), Schnabel (beak), Stuhl (chair), Zigarre (cigar), Schaf (sheep), Panzer (tank), Putz (plaster)* while the abstract paradigm words entail *Fürsorgepflicht (duty of care), erklärlich (accountable), Themenportal (≈thematic portal), Abenteuerlichkeit (adventuresomeness), erfinderisch, (imaginative), Popularität (popularity)* and *verschmähen (to despise sb.).*

Distributional Information: We used the *word2vec toolkit* (see Section 3.1.2) and applied it to a lemmatized version of the DECOW14AX German web corpus (Schäfer and Bildhauer, 2012; Schäfer, 2015). This corpus contains ≈11.6 billion tokens (see Section 3.5 for corpus information). We ignored words that occurred less than 100 times in the corpus. In addition, we tuned the hyper-parameter settings on two German word correlation tasks: *Gur350* (Zesch and Gurevych, 2006) and *Gur65* (Gurevych, 2005). These tasks compare distributional similarity (cosine) with human-annotated similarity values. Finally, we took the vectors that obtained the best performance, in terms of Pearson correlation. The final model used the skip-gram architecture with negative sampling (SGNS) a symmetrical window of size 3, 400 dimensions, and SubSampling with $t = 1e^{-5}$.

Ratings: Using the semantic orientation algorithm together with the word vectors, we were able to obtain 351 617 ratings (86% nouns, 10% verb, 4% adj+adv). After 40 iterations, the final correlation scores between the training and test data are presented in Table 3.5.

	Abs	Ar	Val	Img
Training Correlation	0.838	0.796	0.827	0.832
Test Correlation	0.825	0.784	0.798	0.789

Table 3.5: Final Pearson correlation after 40 Iterations

It can be seen that all the correlations are sufficiently high. Furthermore, the training data provides only slightly higher correlations than the test data. Therefore, we can assume that the algorithm delivered reliable ratings even for words that were not used for the training. Table 3.6 lists some words together with their respective rating score. For each rating type and word class, we present two words with high rating scores and two words with a low rating score.

When all the ratings are compaired pairwise with each other, we observe a high

↑↓	Adj+Adv	*English*	Rating	Verb	*English*	Rating	NN	*English*	Rating
Abstractness-Concreteness									
↑	aufgeblättert	exfoliated	8.44	entlangrutschen	slip along	7.97	Uniformtasche	uniform bag	10.00
↑	beinlang	leg-length	8.32	beklecksen	blot	7.92	Vampirgebiss	vampire ivories	9.61
↓	paradox	paradox	0.63	negieren	negate	0.77	Selbstläuterung	self purification	0.52
↓	rechtfertigbar	justifiable	0.36	innewohnen	inhere	0.64	Willenlosigkeit	abulia	0.66
Arousal									
↑	bestialisch	brutish	9.33	vergewaltigen	rape	9.85	Bandenrivalität	gang rivalry	10.0
↑	gewalttätig	violent	9.25	umbringen	kill	9.32	Blutbad	bloodbath	9.83
↓	satzweise	blockwise	0.81	flechten	weave	1.35	Wortfamilie	word family	0.48
↓	ausgerollt	rolled out	0.71	einfüllen	pour in	1.22	Holzdeckel	wood cover	0.00
Imageability									
↑	neonerleuchtet	neon-lighted	9.83	weiterschnüffeln	continue sniffing	8.99	Granatangriff	grenade attack	10.0
↑	schwarzverhüllt	black-cloaked	9.61	emporzüngeln	upwardslicking	8.71	Polizeijeep	police Jeep	0.87
↓	gewiss	certain	0.58	auffassen	understand	1.08	Zielstellung	objective	0.59
↓	unstrittig	indisputable	0.41	elaborieren	elaborate	1.08	Sinnfreiheit	mindless	0.46
Valence									
↑	wundervoll	wonderful	9.69	beschenken	endow	8.35	fitnessangebot	Fitness offer	10.0
↑	wunderbar	marvelous	9.69	genießen	enjoy	8.28	Frühlingsrezept	spring recipe	9.49
↓	katastrophenmässig	disastrous	0.39	zermürben	demoralize	0.60	Falschdiagnose	misdiagnosis	0.06
↓	ausblutend	bleeding to death	0.37	frikassieren	fricassee	0.57	Essensentzug	food deprivation	0.00

Table 3.6: Example words with high and low ratings per rating type and word class.

Pearson correlation (0.81) between imageability and concreteness and a moderate negative correlation (-0.29) between arousal and abstractness and similarly between valence and abstractness (-0.34). While all other pairwise comparisons exhibit no correlations.

3.4.2 Using Regression

The Semantic Orientation algorithm, shown in the previous section, has many nice properties. Beyond learning ratings for unseen words, one can also look into the different paradigm lists and retrace the algorithm's decisions based on the individual similarities. Hence, the algorithm is very transparent. However, the search for paradigm words takes a lot of time, since in each iteration step the next paradigm word can be each word in the entire vocabulary. When running the algorithm for many iterations using huge vocabularies the algorithm requires a lot of time. Then again, the literature contains different methods used to extend the abstractness norms based on low-dimensional word embeddings. For example, Tsvetkov et al. (2013, 2014) feed word representations into a linear regression classifier. Since there was no comparison of the methods for the supervised learning of the affective norms, we conducted another experiment on the English ratings where we compared different methods (Köper and Schulte im Walde, 2017c).

To ensure reproducibility and to investigate the impact of the underlying word representations, we compared the approaches across different publicly available vector representations[7]; to study the potential differences across vector dimensionality, we compared vectors between 50 and 300 dimensions. The Glove vectors (Pennington et al., 2014) have been trained on 6 billion tokens of Wikipedia plus Gigaword (V=400K), while the word2vec cbow model (Mikolov et al., 2013c) was trained on a Google internal news corpus with 100billion tokens (V=3million).

For training and testing, we relied on the English ratings from Brysbaert et al. (2014), since these ratings are much larger than the available German ratings. Dividing the ratings into 20% test (7 990) and 80% training (31 964) for tuning hyper parameters we took 1 000 ratings from the training data. We kept the ratio between the word classes. The evaluation was done by comparing the new created ratings against the test (gold) ratings using Spearman's rank-order correlation. First, we reimplemented the algorithm from Turney and Littman (2003) (`T&L 03`). Inspired by the recent findings of Gupta et al. (2015) we applied the hypothesis that distributional vectors implicitly encode attributes such as abstractness and directly feed the vector representation of a word into a machine learning classifier, either by using linear regression (`L-Reg`), a regression forest (`Reg-F`) or a fully connected feed forward neural network with up to two hidden layers (`NN`).[8]

	T&L 03	L-Reg.	Reg-F.	NN
Glove50	.76	.76	.78	**.79**
Glove100	.80	.79	.79	**.85**
Glove200	.78	.78	.76	**.84**
Glove300	.76	.78	.74	**.85**
W2V300	.83	.84	.79	**.90**

Table 3.7: Spearman's ρ for the test ratings. Comparing representations and regression methods.

Table 3.7 shows clearly that we can learn abstractness ratings with a very high correlation on the test data using the word representations from Google (W2V300) together with a neural network for regression (ρ=.90). Moreover, the `NN` method significantly outperforms all other methods, using Steiger (1980)'s test ($p < 0.001$).

Cross-Lingual Affect Ratings: In Section 3.4.1, we automatically created ratings based on a small seed set of German ratings and the semantic orientation from

[7] `http://nlp.stanford.edu/projects/glove/`
`https://code.google.com/archive/p/word2vec/`

[8] `NN` Implementation based on `https://github.com/amten/NeuralNetwork`

association algorithm.

In more recent work, we incorporated the findings from the systematic comparison on the English data and again created ratings (Köper and Schulte im Walde, 2018). Unlike the previously created ratings, these ratings were created cross-lingually. This second generation of German norms was motivated by the need to access different norms, that are exclusively available for English. In addition we wanted to explore a cross-lingual method that could be applied to other low-resource languages. Thus, we wanted to create a lexicon for more fine-grained affective classes, including emotions. For English, large lexicons contain such information, such as the *NRC Hashtag Emotion Lexicon* (Mohammad and Kiritchenko, 2015).

Instead of relying on machine translation, we exploited the recent advance in cross-lingual word representation learning. We applied a cross-lingual approach (Smith et al., 2017), relying on orthogonal transformations to learn a linear transformation that aligns monolingual vectors from two languages in a single vector space. We took the off-the-shelf word representations[9] for German and English that live in the same semantic space. Our approach could be easily reproduced for any of the 78 available languages. Note, that the available training resources for affective norms and emotion ratings include English words only. Hence, we learned a regression model based on the English data (ratings and vector representation), and applied it afterwards to the entire German data. We relied on the findings from our previous comparison of methods (Köper and Schulte im Walde, 2017c); therefore, we used a feed-forward neural network as a regression model. The network had two hidden layers, each having 200 neurons. At each step, the input of the network is a single word representation (300 dimensions) and the output is one numerical value trained to correspond to the human annotated (gold) rating for the given input word. After the training, we applied the model to predict a rating score for every word representation in our German distributional space. Since the German representations live in the same space as the English ones, the model was able to predict ratings even for German words based on their vector representation.

The procedure was applied to a range of affective norm datasets. For the training, we took the emotional ratings for 17*k* words from the *NRC Hashtag Emotion Lexicon*; we used *anger*, *disgust*, *fear*, *joy*, and *sadness*. For *valence* and *arousal*, we took the 14*k* ratings from Warriner et al. (2013). For *concreteness*, we relied again on the 40*k* ratings from Brysbaert et al. (2014). Finally, we used the 10*k* ratings for *happiness*

[9] `https://github.com/Babylonpartners/fastText_multilingual`

from Dodds et al. (2011).

Training was always done for each affective norm or emotion in isolation. In total, we obtained nine affective/emotion categories and predicted each to a vocabulary of 2.2 million German words.

3.5 Corpora

In this section, we will describe the underlying corpora used for our experiments. Corpora represent large collections of text, usually taken from the web via web crawling techniques. These texts are usually in a machine-readable format and are often annotated for part-of-speech, lemma and syntactic information. Corpora play a significant role in natural language processing, in particular distributional semantics (see Section 3.1) where the meaning of a word depends on the context in which it occurs, is indeed corpus-based.

For most of our experiments, we rely on the German web corpus **DECOW14AX** Schäfer and Bildhauer (2012); Schäfer (2015). This corpus contains 624 million sentences and ≈11.6 billion tokens. Starting with the publicly available version of DECOW14AX, we gained additional morphology information by applying *SMOR* (Faaß et al., 2010) and *MarMoT* (Müller et al., 2013). For dependency information, we parsed each sentence using the *MATE* dependency parser (Bohnet, 2010). This version of the DECOW14AX corpus was later released as *DECOW16*[10].

SDeWaC (The Stuttgart DEWAC), created by Faaß and Eckart (2013), is a comparably smaller but less noisy web corpus. SdeWaC is a subset of the DeWaC corpus (Baroni and Kilgarriff, 2006), a collection of documents from the .de domain. Each sentence in DeWaC was parsed with *FSPar* (Schiehlen, 2003). The final collection contains only sentences with a low error rate (parsable sentences). SDeWaC contains 44million sentences and 846 million tokens. We are using version 3 of the SDeWac; this version was additionally parsed using the dependency parser from Bohnet (2010). Parts-of-Speech and lemma information comes from the *tree-tagger* (Schmid, 1994).

[10] `http://corporafromtheweb.org/decow16/`

3.5.1 Reconstruction of separated PVs

BVs and particles of German PVs often occur separated over potentially long distances (see Section 2.1.1). Therefore, the correct detection of these PVs is an important preliminary procedure for computational approaches relying on large corpus data. Both of the corpora are provided with part-of-speech and dependency information, using state of the art parsing tools. We, therefore, rely on the procedure from Bott and Schulte im Walde (2014b) to detect separated PVs and to re-attach the particle to the BV.

ID	Wform	Lemma	POS	Dep	DepRel
1	Ein	ein	ART	2	NK
2	Mann	Mann	NN	3	SB
3	**fiel**	fallen	**VVFIN**	0	-
4	heute	heute	ADV	3	MO
5	Nacht	Nacht	NN	3	SB
6	in	in	APPR	5	MO
7	der	der	ART	8	NK
8	Ostseestraße	Ostseestraße	NN	6	NK
9	der	der	ART	10	NK
10	Polizei	Polizei	NN	3	AG
11	**auf**	auf	**PTKVZ**	**3**	SVP

→

Reconstructed
Ein
Mann
auffallen
heute
Nacht
in
der
Ostseestraße
der
Polizei

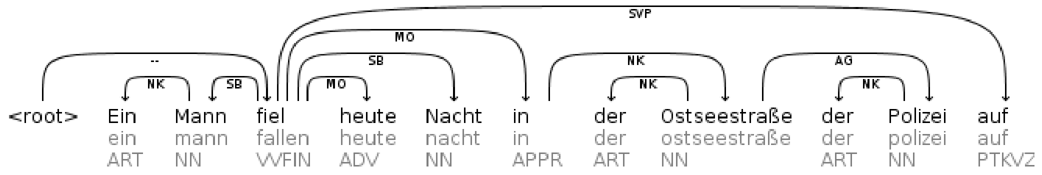


Figure 3.13: Reconstruction of separated PVs according to Parse Information. The original sentence is shown at the left with its dependency structure below. The new reconstructed PV and the resulting sentence is shown at the right.

Figure 3.13 shows an example sentence together with lemma, POS, and dependency information. In this example, the particle-verb "auffallen" *(to strike/attract attention)* is separated. The approach looks for particles by searching for the corresponding tag PTKVZ, which refers to a separable prefix according to the German STTS (Stuttgart–Tubingen Tagset) part-of-speech tagset. Subsequently, by looking at the dependency value (=3 in the example) we obtain the aligned BV fiel/fallen *(to fall)*. Additionally, we check if the BV is annotated with a verb POS tag. To reconstruct the PV, we replace the BV by the concatenation of its lemma and the particle. Then, the particle at its original position, is removed from the sentence.

This approach depends strongly on the quality of the underlying parser and can also lead to false re-attachments. False reconstructions are in particular the case when prepositions are mistaken for particles or vice versa. Typical problems are shown in the sentences a) to c). The combination with German auxiliary verbs (e.g., "haben") are often not annotated correctly as seen in sentence a); here, the particle "vor" is erroneously tagged with APPR (preposition). In contrast, sentence b) without an auxiliary verb is tagged correctly.

a) "Wir haben heute etwas **vor**$_{\text{APPR}}$." (PV: vorhaben)
(We plan something for today)

b) "Wir singen heute etwas **vor**$_{\text{PTKVZ}}$." (PV: vorsingen)
(We sing something today for sb.)

c) "Ich nehme eher **zu**$_{\text{APPR}}$ statt **ab**$_{\text{PTKVZ}}$." (PV: zunehmen & abnehmen)
(I tend to gain weight instead of losing it)

In addition, the method does not account for coordinated particles where basically two different lemmas are seen, as sentence c) shows. Here, the parser tags only "ab" with PTKVZ and not "zu".

It is possible to extend this approach by relying on predefined lookup lists or minimum corpus frequency thresholds. Such heuristics or lists could restrict the set of particles, BVs, or one could restrict the set of possible PV constructions, as was done by Volk et al. (2016) and Batinić and Schmidt (2017). On the other hand PVs are very productive; hence, applying such a restriction can remove infrequent PVs and will ignore neologisms.

3.5.2 Corpus-based Statistics

Despite the rich and mostly theoretical literature on the phenomena (see Section 2.1.2), there is almost no work on the usage of PVs that provides a basic understanding of the phenomena in terms of corpus statistics. While there have been at least three corpus studies for English PVs (Biber et al., 1999; Gardner and Davies, 2007; Liu, 2011), the only work on German PVs in terms of corpus statistics is the work from Khvtisavrishvili et al. (2015), which investigates the frequency proportions in syntactically separated vs. non-separated forms.

Therefore, this subsection provides a corpus-based study of German PVs. We present attempts to identify frequency and usage patterns of German particles. In addition, we look at particle-verb separability and sense information. For our study, we rely on the two large web corpora (see Section 3.5).

We identified particles by looking for the PTKVZ POS tag. Syntactically separated PVs were detected by applying the technique introduced in Section 3.5.1. Since the reconstruction is likely to add noise due to parsing errors, we applied filtering steps: to avoid re-attachment of non-existing BVs, we followed the recommendations from Volk et al. (2016) and restricted the set of BVs to a predefined list of the 5000 most common BVs. In addition we used a min-frequency threshold of 100 for DECOW and 10 for SDEWAC.

Particle Frequency

First, we present the 20 most frequent particles, as identified by our web-corpus.

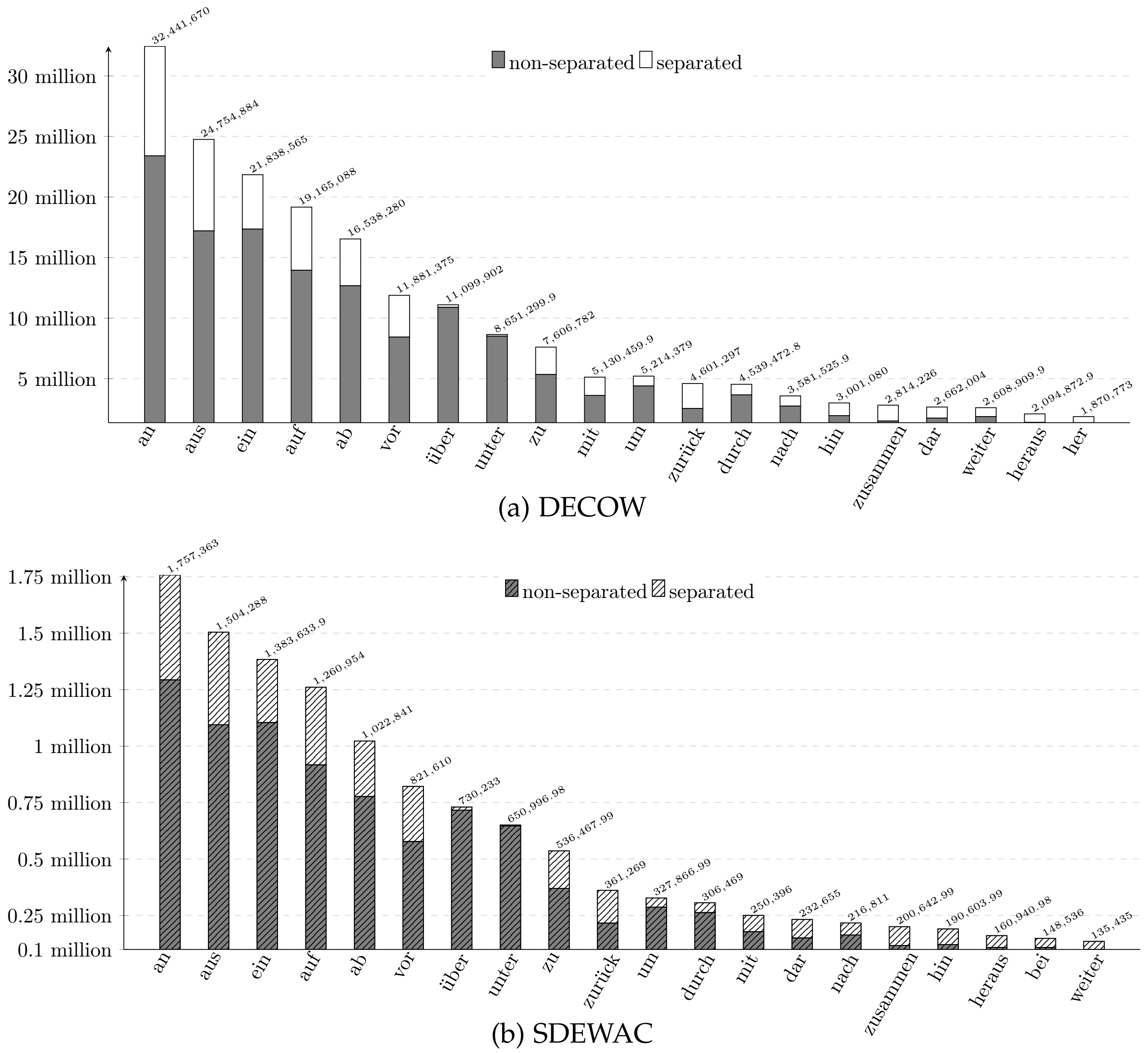
The first nine particles are identical across corpus with a striking similarity with respect to separability. In addition, the distribution exhibits an almost Zipf-like behavior where only a few particles are highly frequent and the remaining ones are less frequent. Our approach cannot distinguish between prefix verbs and PVs; hence, "durch", "über", "um" and "unter" can contain (non-separable) prefix verbs which might explain their low percentage of syntactically separated occurrences.

Particle Separability

Figure 3.15 shows the separability distribution per particle. The underlying information for this plot is a list of all PVs, together with their probability of appearing syntactically separated or non-separated. For example the PV "abfliegen" *(to depart)* appears in 40% of its corpus occurrences separated and in the remaining 60% non-separated. Unlike Figure 3.14, this plot gives equal weight to each PV and ignores the individual frequency of the respective verb.

We removed "über" and "unter", since their lists contained many prefix verbs and, additionally, we removed "dar" because the set of PVs for this particle was too small to draw reliable conclusions from its distribution.

Overall, the figure shows how many verbs Y (in percentage, Y-Axis) have a separability chance of X (also in percentage, X-Axis). Figure 3.14 indicated already that most PV corpus occurrences are observed for the non-separated case. Figure 3.15



(a) DECOW

(b) SDEWAC

Figure 3.14: Frequency Distribution: 20 Most Frequent Particles across two Corpora. (Note: Y-Axis are different scaled)

compares this pattern. It can be seen that most PVs have a low separability-chance.

Furthermore, it can be seen that the four most frequent particles, namely "ab", "an", "auf" and "aus" exhibit very similar distributions. For these four particles the largest bar consists of 25-30% of all PVs, which appear separated from their BVs in only 10% of their corpus occurrences.

A manual inspection revealed that most border cases, namely the ones that are always separated or never separated, are likely parsing errors and, therefore, are not PVs. We did not address further experiments to estimate the amount of false reconstructions, which is beyond the scope of this thesis. Furthermore, heuristics and

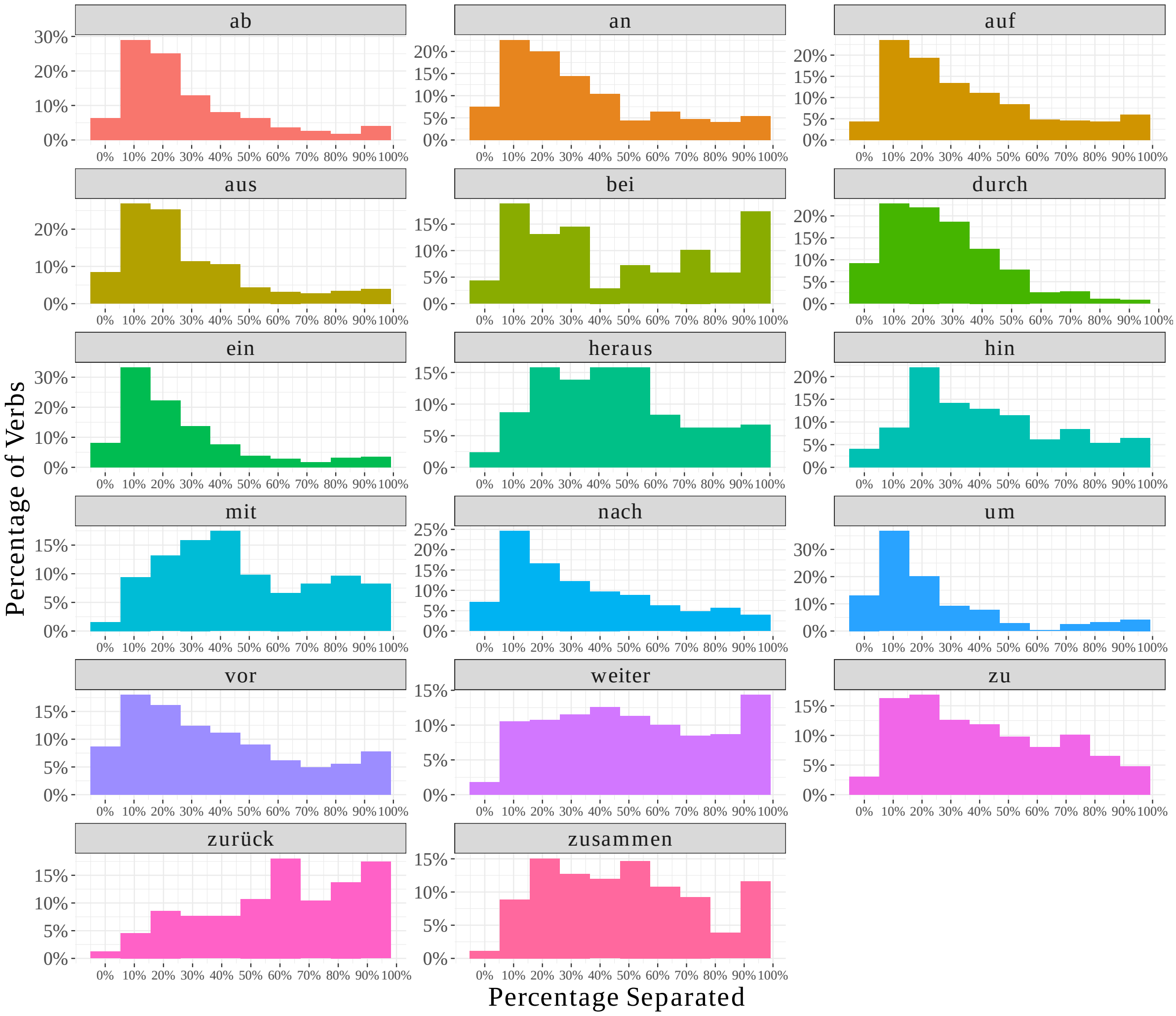


Figure 3.15: Histogram of Separability per Particle Type. Counts are divided into 10 bins. Corpus: DECOW. (Note: Y-Axis is scales different for each Particle)

thresholds could be used to exclude such cases. The correct detection of separable PVs is still an active and difficult research question. For that reason such figures, as the one presented above, can benefit from more sophisticated methods to detect separated PVs.

Distance: Another interesting phenomenon, which, to the best of our knowledge, has not been examined before now, is the distance between PVs and BVs. Across PVs, with a min frequency of 100, we counted the average distance based on the tokens between BV and PV. Using this information, we computed a violin plot (Figure 3.16) across the entire data based on DECOW14. From all the verb samples the probability density function is estimated and is depicted along the X-Axis. If the violin is wide at a certain location, it means this distance is especially likely. We divide the violin into the quantiles of 25, 50, and 75. It can be seen that 50% of all verbs have a mean distance within [0,2.5] and 75% fall below a mean of 4.5. Larger distances are comparably unlikely and might be caused due to parsing errors. Interestingly, we found many double-particles with high mean distances. Examples here include vorbei *(vorbeischrammen: to scrape past, vorbeiführen: to go past sth.)*, entgegen *(entgegenziehen: pull against, entgegennehmen to receive)*. High frequency PVs, on the other hand, show a lower mean distance.

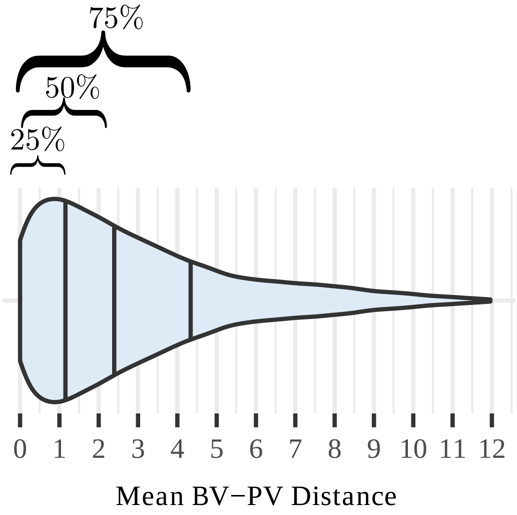
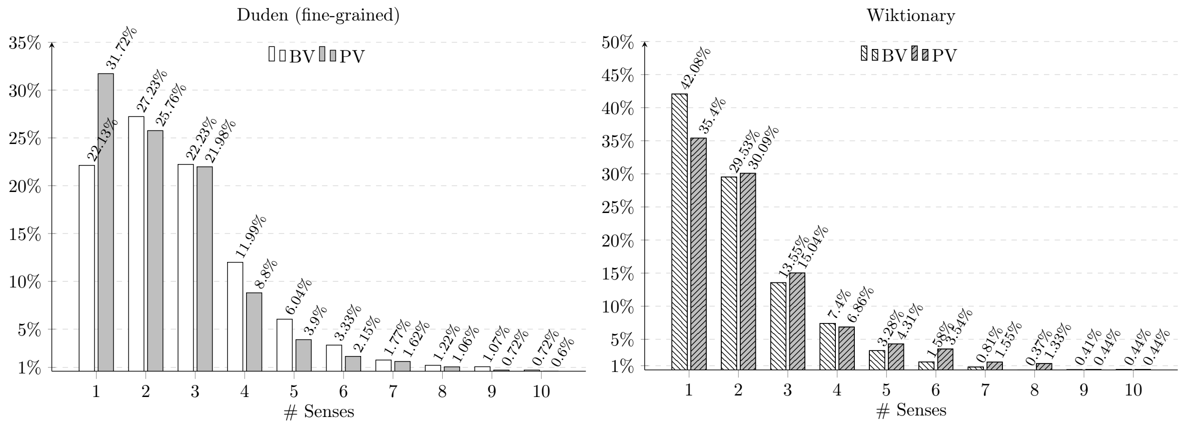


Figure 3.16: Violin plot based on the mean (Token) Distance between Base Verb and Particle Verb.

Base Verb and Particle Verb Senses

We are now addressing the question whether PVs are more ambiguous, i.e., have more senses, than regular verbs do. Although PVs represent a productive phenomenon, we can address this research question by relying on lexicon information. Hence, to obtain sense information, we used two German dictionaries, the Duden resource and the German Wiktionary. For each verb, we counted the number of senses and computed a histogram for base and PVs. Duden provides coarse and fine-grained sense information; we extracted the fine-grained counts. In total, we found 7224 verbs in Duden and 3620 verbs in the German version of Wiktionary.

Figure 3.17 shows the resulting histogram. The X-Axis refers to the number of senses and the Y-Axis shows the percentage of verbs that fall into this category. It



(a) Duden (b) Wiktionary

Figure 3.17: Number of Senses for Base Verbs vs. Particle Verbs based on Duden and Wiktionary

can be seen that across the resources, the majority of verbs for both of the categories are labeled with 1-3 senses. Furthermore, we see no strong difference between the BVs and the PVs. The histogram based on Duden (Figure 3.17a) supports the view that BVs are more likely to appear with many senses. According to Duden, 22% only of the BVs are monosemous in contrast to 31% of all PVs.

Wiktionary on the other hand shows the opposite effect, where PVs are more likely to have many senses.

In summary, this plot shows that there are no strong differences between the number of senses for base and PVs based when looking at sense information from publicly available dictionaries. While this is the case, it must be clear that such resources cover only a fraction of the productive PV phenomena and are biased toward high frequency verbs.

Number of Senses and Corpus Frequency: Combining the sense information from Duden and frequency counts from a large web corpus (DECOW14), we additionally looked at the correlation between frequency and number of senses.

Figure 3.18 shows that the number of senses tends to increase with corpus frequency. Again, we observe similar patterns for BVs and PVs. Within the twenty most frequent PVs, we find many examples with more than 10 senses. High frequency and highly ambiguous PVs include: "ausgehen" *(to end / date / burn out / run dry, ...)* , "aufnehmen" *(to gather, accommodate sb. / ingest / record ,...)*, "einstellen" *(to adjust / hire,stop,..)*, "auftreten" *(to appear / emerge /exist / kick sth. open)*, "vorstellen" (*to*

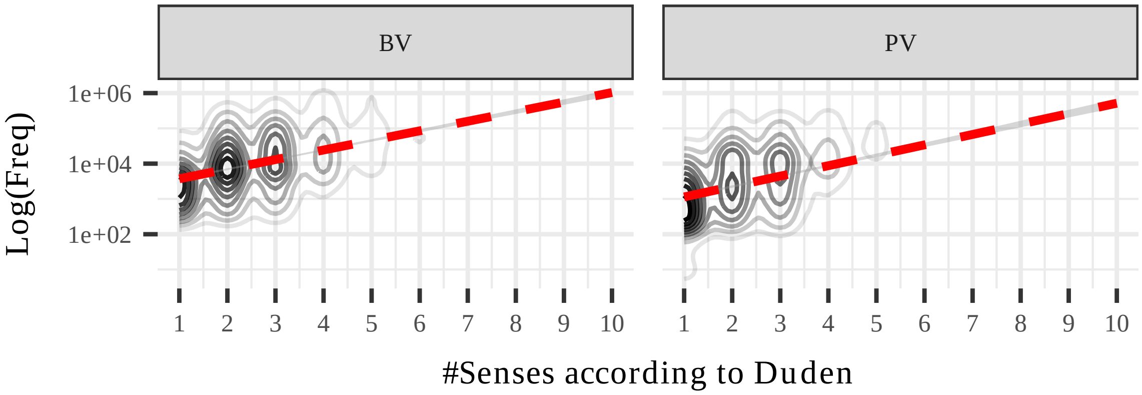


Figure 3.18: Correlation between Verb Frequency and Number of Senses. Red line shows linear fit, circles indicate data density.

introduce / suggest / put in front, ...) and "anbieten" *(to offer / supply / present / serve,...)*.

4 Compositionality

The principle of (semantic) compositionality is that the meaning of the whole (word, phrase) is determined by the meanings of its parts and their combination. Predicting the compositionality of a PV can be seen as a first step toward modeling the meaning of the PV from the meaning of its BV and the particle. Being able to predict the degree of compositionality of unseen PVs is highly desirable, as these multi-words represent a very productive phenomenon (see Section 2.1).

Downstream applications, such as machine translation, information extraction, or question answer systems, could benefit from recognizing opaque PVs that might deserve special attention or treatment. Moreover, identifying important properties that enhance the correct prediction of compositionality are also interesting from a theoretical perspective. Such information can provide insights into the semantics of PVs and the nature of the productive process.

We apply a view on compositionality[1], that assumes that compositionality represents a continuum between entirely compositional or transparent and non-compositional or opaque.

In the following sections, we model particle-verb compositionality. Across experiments, we rely on distributional semantics to approximate meaning. We model compositionality from two different perspectives:

1. Subsection 4.1 compares PVs with their corresponding BVs. We build a computational model that is able to distinguish opaque from compositional PVs in a ranking setup. This setup requires manually collected information.

2. While the first experiment models the similarity between a BV and a PV, Subsection 4.3 presents experiments where we model the contribution of the particle. In this setup, we learn a vector modification for each particle that allows us to synthesize or predict the vector of the resulting PV given a BV and a particle. Unlike the first experiment, this methodology requires no human-labeled data

[1] Related work, in particular on MWEs, contains views where such expressions are seen as entirely idiosyncratic e.g., Sag et al. (2002).

but needs a set of training observations. These observations are, in our case, tuples between the BV and PV vector representations for a given particle.

A subset of the experiments described in this chapter was published in Köper and Schulte im Walde (2016) and Köper et al. (2016).

4.1 Compositionality via Vector Similarity

DSMs (Section 3.1) rely on the *distributional hypothesis* (Harris, 1954), that words with similar distributions have related meanings. They represent a well-established tool for modeling semantic relatedness between words and phrases (Bullinaria and Levy, 2007; Turney and Pantel, 2010). Predicting degrees of PV compositionality using distributional models has been already addressed, mainly for English and German (see Section 2.1.2).

In this subsection, we present and perform experiments using established techniques to model compositionality.

4.1.1 Methodology

The majority of approaches predict compositionality by learning and comparing vector representations for BVs and PVs respectively. For example, the vector of the opaque PV "unterjubeln" *(to plant sth. on sb.)* is compared against its BV "jubeln" *(to cheer)*.

For comparison, a vector distance is applied (e.g., cosine similarity). The estimated similarity value is, then, compared against human ratings of compositionality. A good distributional model should obtain a high correlation with the human ratings (gold standard).

While this setup is a well-established technique to measure compositionality, we want to point out explicitly that this methodology is entirely type-based and, therefore, cannot account for ambiguity. Hence, we are not sure if the compositionality ratings correspond to the predominant word sense as perceived by the rater. On the other hand, psychological studies have shown that humans consider different meanings of a word while judging semantic similarity of word pairs (Tversky, 1977; Markman and Gentner, 1993).

The construction and use of a distributional model involves many design choices (see Section 3.1). As a first experiment towards modeling PV meanings, we, therefore, want to systematically explore a variety of different DSMs to predict compositionality.

DSMs: for the word representation we use embeddings (predict models, Section 3.1.2) based on *word2vec* (Mikolov et al., 2013c), we obtained representations using the skip-gram architecture with negative sampling (set to 15). The representations differ with respect to corpus, window size, and dimensionality. Further parameters were set to their default value. In addition, we explored the impact of the particle-verb reconstruction (Section 3.5.1); thus, we re-attached separated PVs. In all the settings, we relied on lemma instead of word forms. In total, we compared 160 different vector spaces, as Equation 4.1 illustrates:

$$\overset{\textbf{Corpus}\ \textbf{(2)}}{\begin{pmatrix} DECOW14AX \\ SDeWaC \end{pmatrix}} \times \overset{\textbf{Wind.}\ \textbf{(10)}}{\begin{pmatrix} 2 \\ 4 \\ 6 \\ 8 \\ 10 \\ 12 \\ 15 \\ 20 \\ 25 \\ 30 \end{pmatrix}} \times \overset{\textbf{Dim.}\ \textbf{(4)}}{\begin{pmatrix} 50 \\ 100 \\ 200 \\ 300 \end{pmatrix}} \times \overset{\textbf{Mod.}\ \textbf{(2)}}{\begin{pmatrix} \textit{Default} \\ \textit{reconstructed PVs} \end{pmatrix}} = \underline{\textbf{160}} \tag{4.1}$$

Target Gold Standard(s): As a gold resource, we focused on the collection from Bott et al. (2016). Their resource (*GhostPV*) contains a set of 400 PVs across 11 particle types: ab, an, auf, aus, durch, ein, nach, um, unter, zu, and über. Furthermore, the verbs are balanced over three frequency bands (low, middle and high). The compositionality information was collected via the crowdsourcing platform Amazon Mechanical Turk[2](MTurk). On average, 16.14 raters annotated a given BV-PV pair on a scale from 1 (low compositionality) to 7 (high compositionality). The final score for a given item is the mean score of all the available raters.

While our focus relied on the largest collection (*GhostPV*), to make sure that our findings are not artefacts of the gold resource we additionally report results on a smaller collection from Bott and Schulte im Walde (2015). This resource contains 150 BV-PV pairs across the same eleven particle types. We refer to this recourse as

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

PV150[3].

Figure 4.1 shows the histogram based on the gold rating scores for both of the resources. It can be seen that most of the BV-PV pairs obtained a medium compositionality score. Hence, only a small subset of the data falls in the high compositionality and the low compositionality categories.

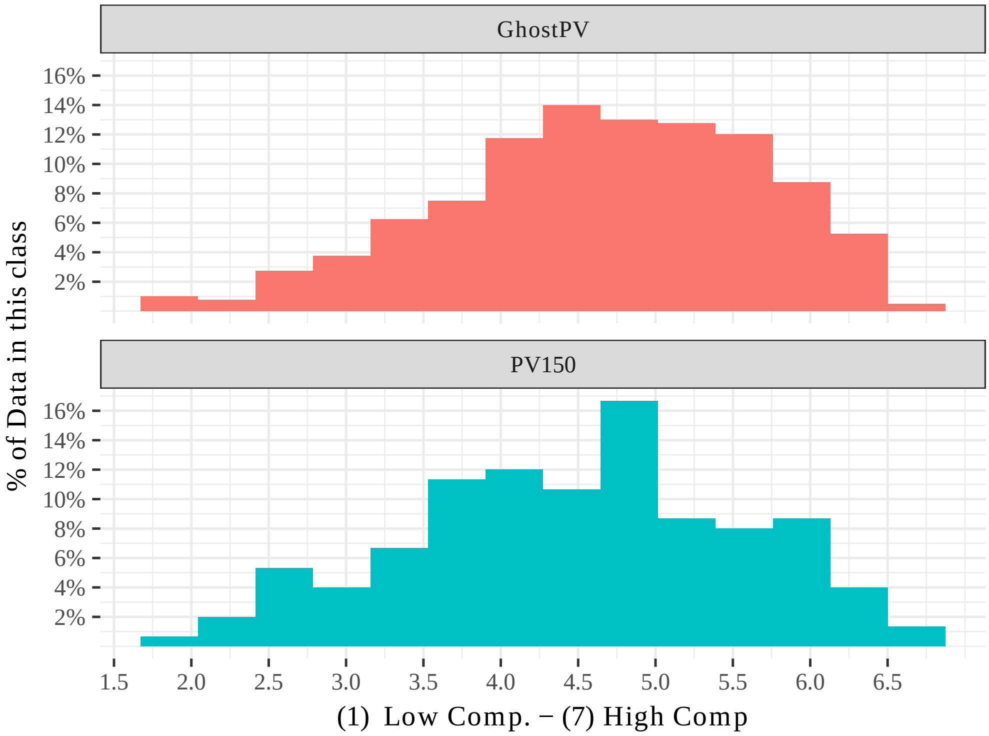


Figure 4.1: Histogram: GhostPV and PV150. Distribution of mean gold compositionality scores. Bins set to 14.

Predicting Compositionality We evaluated each of the 160 distributional vector spaces. The similarity of a PV–BV vector pair as measured by the *cosine-similarity* was taken as the predicted degree of compositionality. The Spearman's rank-order correlation coefficient ρ (Section 3.2.2) is used to compare the overall ranking of pair similarities to the gold standard compositionality ratings. High Spearman values indicate good or more human-like performance.

[3] Bott and Schulte im Walde (2015) refer to this resource as GS3. We changed the name, as *PV150* is less confusing.

4.1.2 Results

Corpus, Window and Dimensionality: We first look at the results with respect to corpus, window, and dimensionality. Hence, we ignore the PV-reconstruction and focus on the 80 default distributional spaces. Figure 4.2 shows Spearman's ρ across these settings.

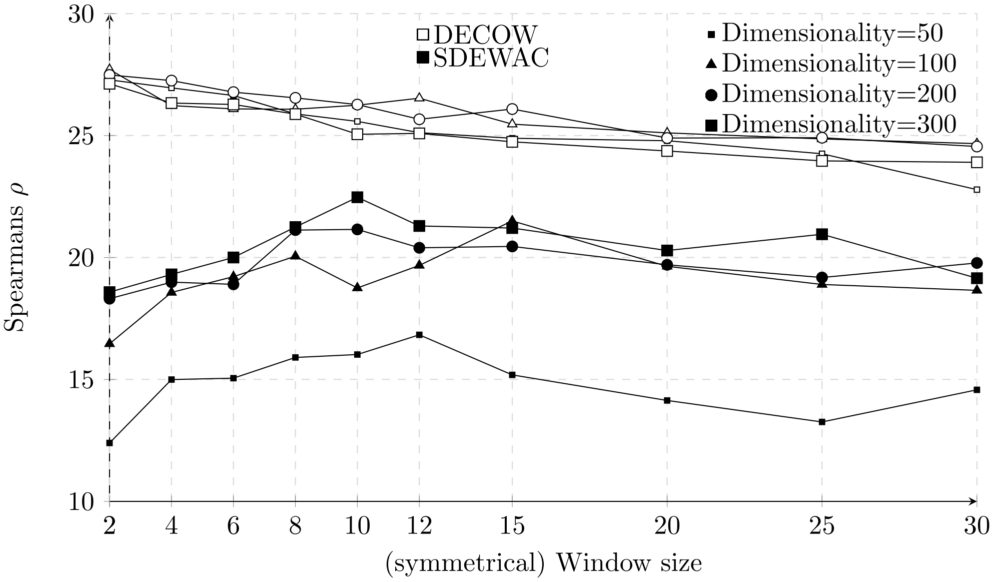


Figure 4.2: Line plot of Spearman's ρ for *GhostPV* across corpus, window, and dimensionality.

Overall, we notice that the performance in terms of Spearman's ρ is comparably low and obtains values below $\rho = 30$. It is likely that the many pairs with a medium compositionality score in the gold standard make this task challenging for a distributional model. In spite of this, we observe a large margin between the worst model (ρ =12.39) and the best model (ρ =27.71).

The almost 15× larger web corpus (DECOW) obtains a clearly superior performance to the smaller SDEWAC corpus across all settings. This indicates that the distributional representations benefit from more corpus occurrences and the increasing amount of contextual information.

In the same vein, larger dimensionality leads to better performance for the SDEWAC representations but has little effect for the DECOW representations. A dimensionality of 50 is not recommended as this parameter performs particularly poorly for the smaller corpus. Interestingly, we observe different preferences with respect to window size. The DECOW representations perform best with a small window (2), whereas the SDEWAC representations obtain better performance with a larger window (= 10 − 12). To confirm that the findings are stable across different compositionality resources, we provide the results for the same setup with the *PV150* gold standard in Figure 4.3. The results across the two different resources are remarkably similar.

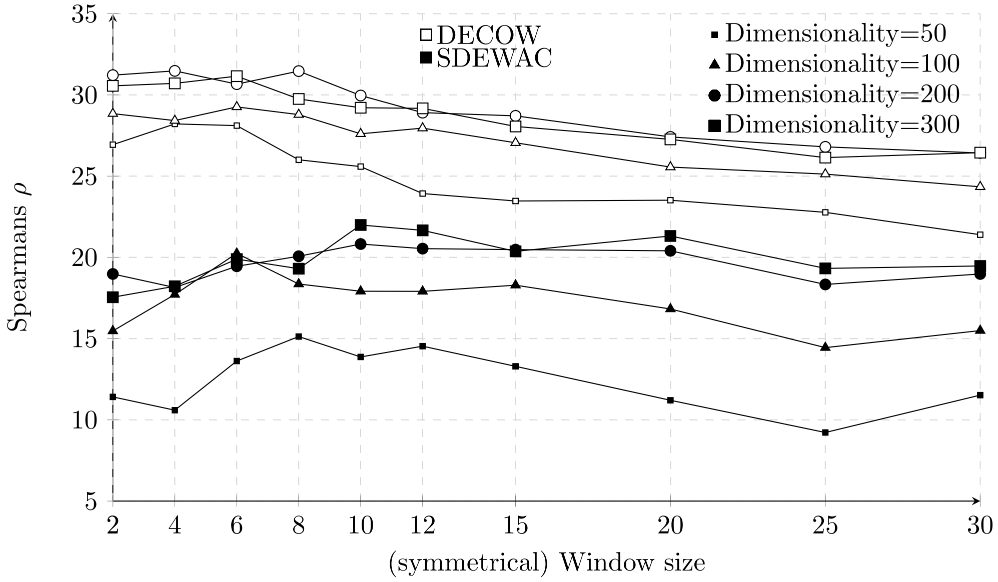


Figure 4.3: Line plot of Spearman's ρ for *PV150* across corpus, window and dimensionality.

PV-Reconstruction: In the following, we describe and discuss our results with respect to the re-attachment of separated PVs, as presented in Section 3.5.1. Recent work from Reimers and Gurevych (2017) show that setups having many hyper-parameters should be evaluated by reporting score distributions to draw correct conclusions. Although they focus on sequence tagging tasks, we acknowledge this idea and we present results as score distributions, i.e., we provide a plot over the distribution for the 80 different models in the default setting compared to the distribution of the 80 models with the reconstruction. Figure 4.4 shows the distribution divided according to the gold standard with and without the usage of PV-reconstruction. We observe a striking increase in performance when re-attaching separated particles to their BVs.

In contrast to the related work, our findings confirm the ones from Bott and Schulte im Walde (2015). Since there are no published results for the *GhostPV*

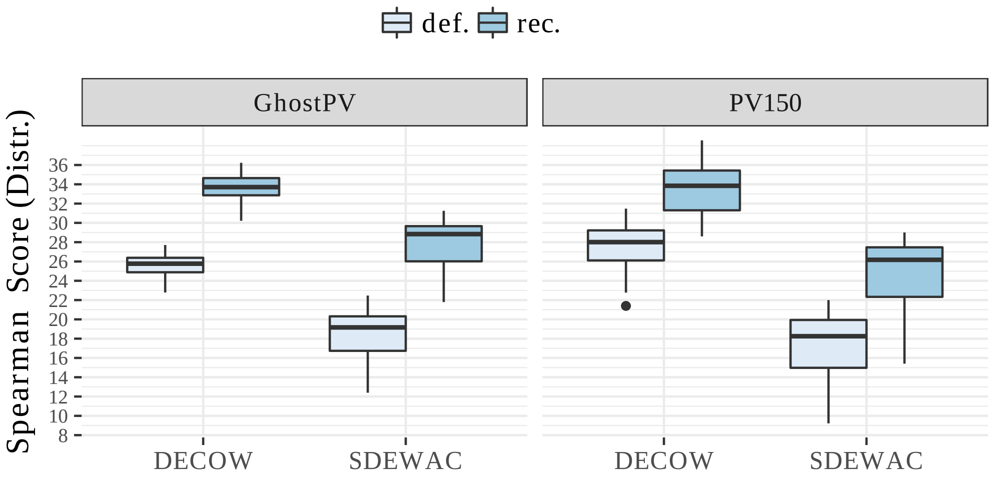


Figure 4.4: Box plot of Spearman's ρ Distribution with and without PV-Reconstruction across tasks

dataset, we restrict this comparison to the *PV150* dataset. Their work is done across count models with their best performing model being one without dimensionality reduction, using a window of size 5 together with LMI (Section 3.3) as feature weighting. They also reattach separated PVs and rely on the SDEWAC corpus, obtaining a score of 28 which is notably similar to our embedding models, the ones using a dimensionality above 50 all obtain scores between $26 - 28$ for windows within the range of 4-6.

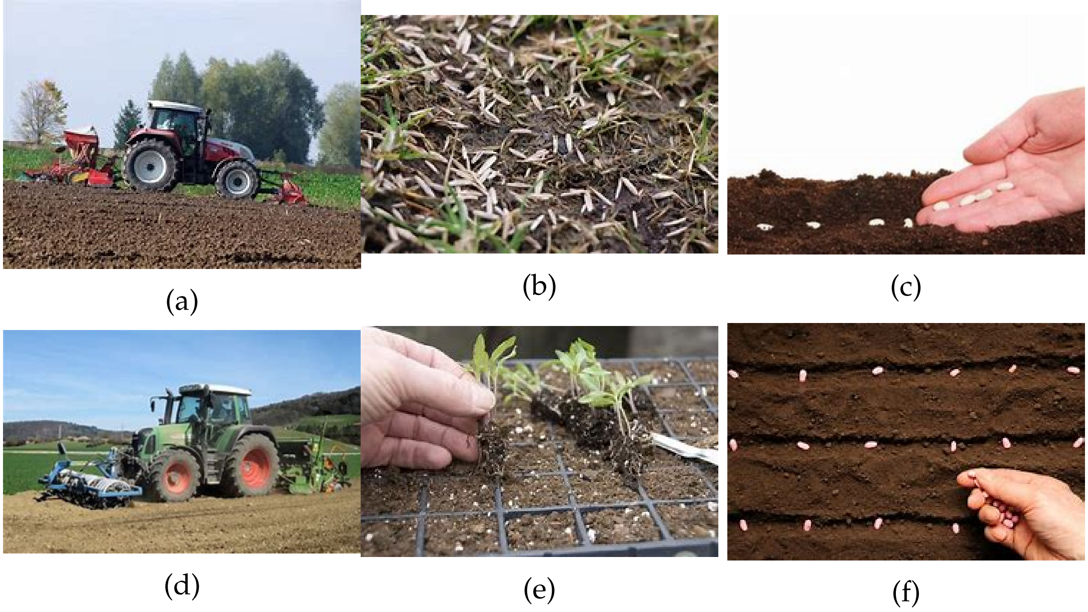
4.2 Compositionality via Multi-Modal Vector Similarity

4.2.1 Introduction

A typical shortcoming of the traditional distributional models is that they approximate meaning entirely in terms of other words without links to other modalities or the outside world (perception). This issue is often referred to as the symbol grounding problem (Harnad, 1990).

Therefore, in the last few decades, standard DSMs using bag-of-words or syntactic co-occurrence counts have been enhanced by integrating perceptual information (Silberer and Lapata, 2014; Bruni et al., 2014; Roller and Schulte im Walde, 2013; Kiela et al., 2014; Lazaridou et al., 2015b).

With respect to PV compositionality, visual information can be used to distinguish transparent from opaque constructions. In Figure 4.5, we see that some of the images obtained from a search engine (bing.de) for the highly compositional BV-PV combination "säen-aussäen" are almost identical.


(a) (b) (c)

(d) (e) (f)

Figure 4.5: Three examples taken from the top 25 image results from bing to illustrate visual similarity. Top shows säen (a-c) and bottom shows aussäen (d-f)
(both verbs can be translated as "to sow")

While standard DSMs have been applied to a variety of semantic relatedness tasks such as word sense discrimination, selectional preferences, and relation distinction (among others), multi-modal models have predominantly been evaluated on their general ability to model semantic similarity as captured by *SimLex* (Hill et al., 2015), *WordSim* (Finkelstein et al., 2002) and so on,. In addition, most of the work focuses clearly on nouns.

We are now going to extend the previously introduced standard DSM that relies on textual co-occurrences (Section 4.1) with a multi-modal model extension that integrates visual information. Thus, differently to most of the previous multi-modal approaches, we address verbs and a semantically specific task. Furthermore, we investigate the use of methods to increase the quality of predictions, such as filters to optimize the visual space, relying on dispersion and imageability filters (Kiela et al., 2014), and a novel clustering filter.

4.2.2 Multi-Modal Vector-Space Models

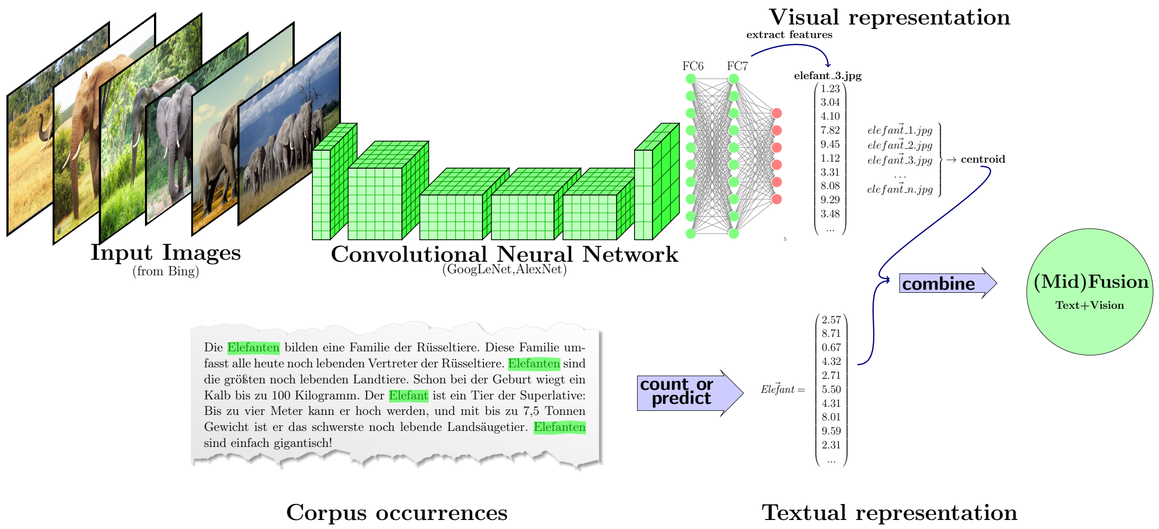
The visual features rely on images downloaded from the *bing* search engine during October 2016 using the (now outdated) bing custom search API. The earlier work on multi-modal approaches relied often on feature norms (Silberer and Lapata, 2012; Hill and Korhonen, 2014) or on raw image data. When using raw image data, the representations were often based on the *bag of visual words* (BoVW) approach (Bruni et al., 2014; Roller and Schulte im Walde, 2013; Kiela et al., 2014). Recently, more attention is being spent on the usage of neural network architectures, that were designed for large-scale image classification, to obtain low-dimensional dense numerical representations of images. The comparison by Kiela and Bottou (2014) showed a clear performance gain over features based on the traditional BoVW approach. Hence, following the recommendations of Kiela et al. (2016), we queried 25 images per word, and converted all images into high-dimensional numerical representations by using the caffe toolkit (Jia et al., 2014) and pre-trained (convolutional) neural network models (CNN)).

In the default setting, a word is represented in the visual space by the mean vector of its 25 image representations. As image-recognition neural network models, we used:

1. GoogLeNet (Szegedy et al., 2015), a 22-layer deep network. It obtained state of the art results on the ImageNet Large-Scale Visual Recognition Challenge 2014. We obtained the vectors by using the representation of layer $pool5/7x7_{s1}$, i.e., the last layer before the final softmax, containing 1024 elements (= dimensionality).

2. AlexNet (Krizhevsky et al., 2012), a neural network with five convolutional layers (4096 dimensions). We used the vectors from layer $fc7$, a 4096-dimensional feature vector for every image.

Figure 4.6 illustrates the process. The neural network architecture in this figure is AlexNet, with the first layers being convolutional layers and the last ones being fully connected layers. There are several possible ways of combining or fusing textual and visual information. These categories are divided according to the stage of combination: early-fusion refers to jointly learning the representations. Mid-fusion refers to learning them independently and combining them before computing similarity scores. Late fusion, on the other hand, learns representations independently and computes similarity scores independently and then combines the scores.



Die Elefanten bilden eine Familie der Rüsseltiere. Diese Familie umfasst alle heute noch lebenden Vertreter der Rüsseltiere. Elefanten sind die größten noch lebenden Landtiere. Schon bei der Geburt wiegt ein Kalb bis zu 100 Kilogramm. Der Elefant ist ein Tier der Superlative: Bis zu vier Meter kann er hoch werden, und mit bis zu 7,5 Tonnen Gewicht ist er das schwerste noch lebende Landsäugetier. Elefanten sind einfach gigantisch!

Figure 4.6: Pipeline, creating a multi-modal vector space model. Example, obtaining a multi-modal vector for "Elefant".

Theoretically, early-fusion is the most interesting as it aims to learn both representations jointly, just like humans hear words accompanied with visual stimuli. However, early-fusion has several drawbacks, such as requiring joint data and a sophisticated model to learn based on both modalities. Since we work with different data sources, we relied on mid-fusion, i.e., the concatenation of the L2-normalized representations (Bruni et al., 2014). Note that normalization is performed before the concatenation of both modalities. This ensures that visual modality and textual modality contribute equally to the overall multi-modal representation[4].

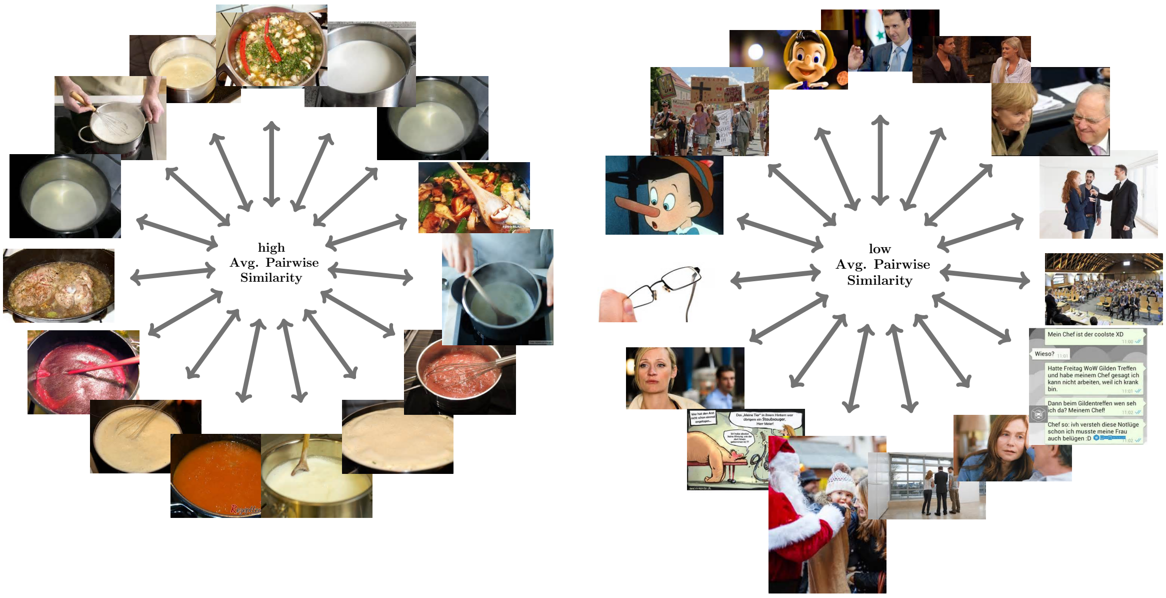
4.2.3 Visual Filters

In our experiments, we compare the predictions of compositionality across all the targets in the gold standards. Furthermore, we explore various optimizations of the visual space. In accordance with human concept processing (Paivio, 1990), including image representations should be more useful for words that are actually visual. We experiment with the following three strategies to enhance our multi-modal space:

1. *Dispersion-based filter* suggested by Kiela et al. (2014). The filter decides whether to include perceptual information for a specific word or not, relying on a pair-

[4] We later rely on the cosine similarity as a distance measure, which is the normalized dot-product. Since our modalities are normalized before concatenating, we could formulate this also as a late-fusion strategy that gives equal weight to both modalities.

wise similarity between all the images of a concept. This is illustrated in Figure 4.7a and Figure 4.7b, where the pairwise similarity is very high and respectively low.



Mein Chef ist der coolste XD 11:00

Wieso? 11:01

Hatte Freitag WoW Gilden Treffen und habe meinem Chef gesagt ich kann nicht arbeiten, weil ich krank bin. 11:01

Dann beim Gildentreffen wen seh ich da? Meinem Chef! 11:02

Chef so: ivh versteh diese Notlüge schon ich musste meine Frau auch belügen :D 11:02

(a) High Pairwise-Similarity.
Example images for aufkochen *(to boil up)*.

(b) Low Pairwise-Similarity.
Example images for anlügen *(to lie to sb.)*.

Figure 4.7: Illustration of dispersion (filter)

The underlying idea is that highly visual concepts are represented by similar pictures and, thus, trigger a high average similarity between the word's images. Abstract concepts, on the other hand, are expected to provide a lower dispersion. For a given word, the filter decides whether to use only the textual representation, or both the textual and visual representations, depending on the dispersion value. This method requires no external information about imageability but requires a predefined threshold (set to the median of all the dispersion values).

2. We apply an *rating filter* based on external imageability norms that were automatically extended to larger vocabularies (Section 3.4.1, the norms created by Köper and Schulte im Walde (2016)), are used to successively include only images for the most imaginable target words. This filter is applied in the same way as dispersion relying on a threshold (median).

3. We suggest a novel *clustering filter* that performs a clustering of the 25 images for a given concept (verb). The clustering operates only on a small set of elements, hence we rely on the SEMCLU (SEMantic CLUstering) algorithm with dynamic thresholding from Apidianaki (2010). By default, this method does not take imageability into account but is designed to obtain a purer visual representation that focuses on the predominant sense of the available images.

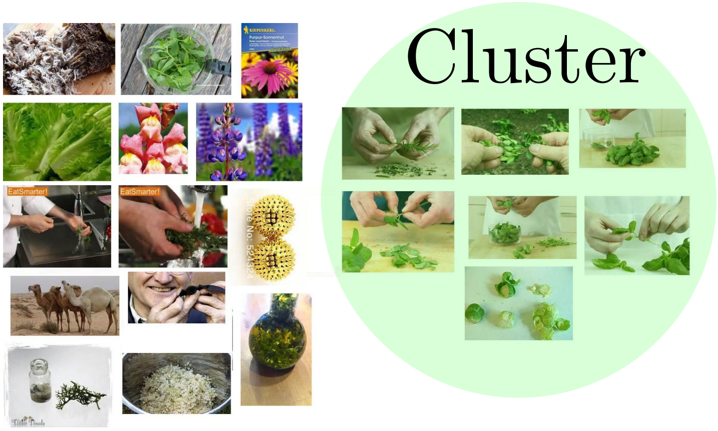


Figure 4.8: Largest Image cluster (right), based on the 25 Images for "abzupfen" *(to pick sth.).*

The clustering is performed in a bottom-up manner, whereby in the first step two images with a high similarity combine to form an initial cluster. In the second step, the two-image clusters may be enriched by additional images, by applying a recursive function to enrich clusters based on pairwise similarity and an image specific dynamic threshold. The algorithm is applied until convergence for every concept, with each having at maximum 25 images. This leads to clusters with very high pairwise similarity. After the clustering we rely only on the images from the largest resulting cluster and remove the remaining images.

In Figure 4.8 we show the resulting largest cluster (shown right) in contrast to the other images. The other images were merged in the remaining clusters (shown left). It can be seen that this algorithm filters unwanted or noisy images.

4.2.4 Results

We extended the previous setup, using 160 different vector-space models, two compositionality gold-standard resources, and two different neural network architectures (AlexNet, GoogLeNet) with our filtering methods. Therefore, we investigated four different modifications: the three visual filters (dispersion, ratings, clustering) as well as a default setup where we apply no filtering and always integrate the unmodified images via mid-fusion (AlwaysBoth). We report results again in terms of score distributions over Spearman's ρ.

Visual Filters Results: The results (using GoogLeNet) for both of the gold standards are shown in Figure 4.9 for GhostPV and in Figure 4.10 for PV150. We divide the plots according to default and reconstruction, as we have seen already that this modification results in a larger margin.

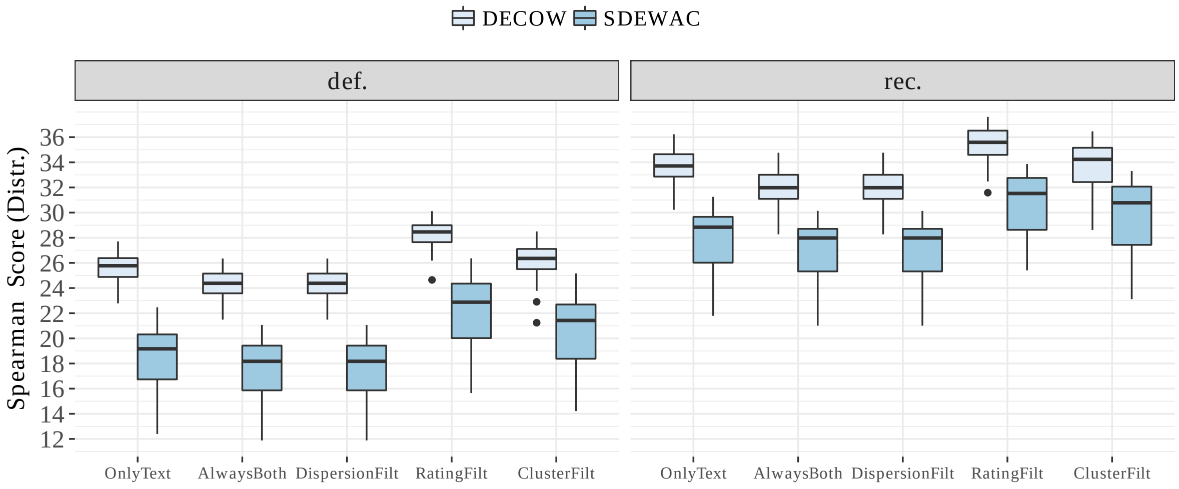


Figure 4.9: Comparison of textual (only) and various multi-modal spaces for **GhostPV**. Distribution is computed across corpus, window, dimensionality. Visual information based on GoogLeNet.

The resulting pattern is stable across gold standard and setup. We obtain a visible improvement in the multi-modal model, but only when applying the rating based filtering or the cluster filters. For that reason we also investigated the combination of rating and cluster filter (not shown in the Figure). The effect is a performance more or less identical to the rating filter. In the other cases, the visual modality tends to decrease the performance slightly. Another essential point is the performance of the

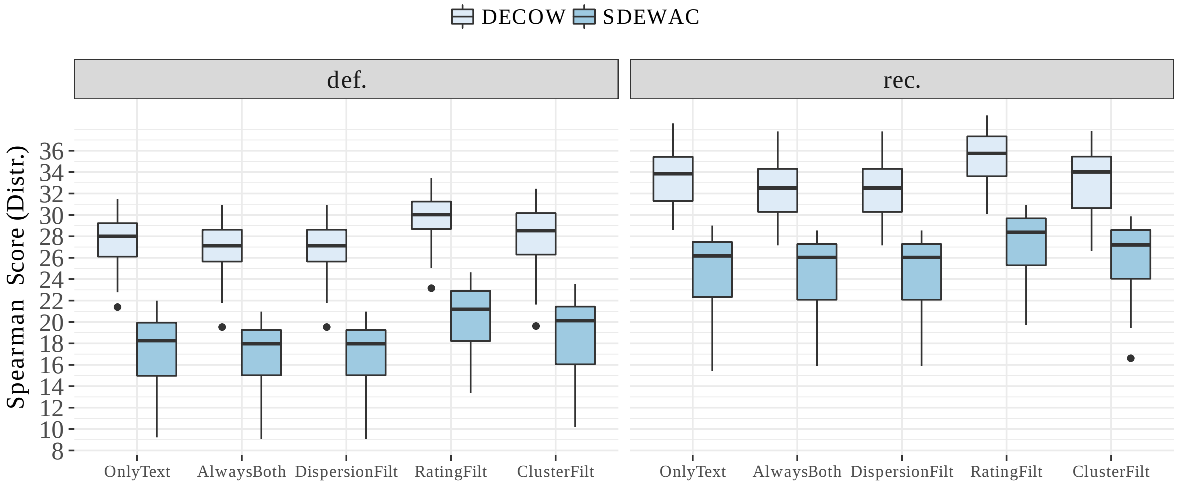


Figure 4.10: Comparison of Textual (only) and various Multi-Modal Spaces for **PV150**. Visual information based on GoogLeNet.

visual representation in isolation (OnlyVis) which is not shown, as this performance was close to zero. Both neural network architectures obtained a score only within $[3, 5]$ points in ρ when evaluated in isolation. This indicates that we cannot rely on the visual modality without textual information to predict compositionality.

CNN Architecture: We explored two neural network architectures, AlexNet and GoogLeNet. In our multi-modal evaluation on the task of compositionality predictions, we found that both architectures perform very similarly. We noticed only marginal differences with respect to the neural network architecture, confirming the findings of Kiela et al. (2016). Figure 4.11 shows score distributions for the multi-modal methods across both tasks, and the 160 different semantic spaces. That aside, GoogLeNet requires less memory since it occupies only 1024 dimensions per image in contrast to the 4096 of AlexNet, making GoogLeNet the more suitable architecture in our setup.

4.2.5 In-Depth Analysis of a Single Vector Space

Performance on Target Subsets: A common technique to explore the influence of constituent properties, is to divide the data according to the properties of interests (Bott and Schulte im Walde, 2014b; Schulte im Walde et al., 2016). However, this is problematic; it results in a new gold standard per property, which is potentially very

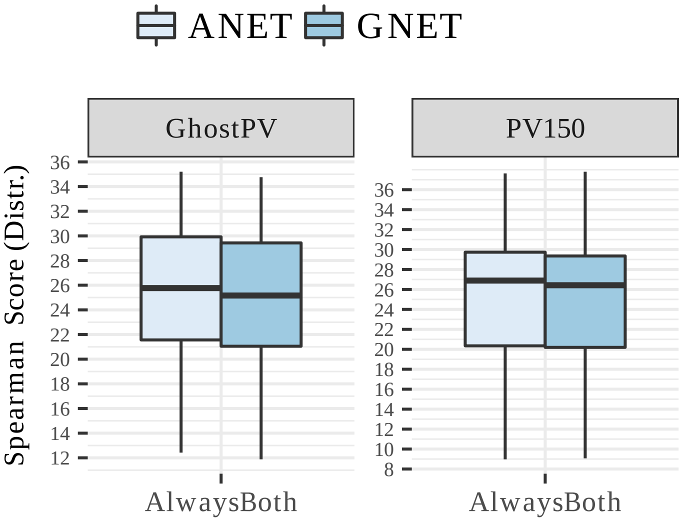


Figure 4.11: Score Distribution GoogLeNet (GNET) vs. AlexNet (ANET) across Tasks.

different and not directly comparable with the original resource.

For this analysis, we focus on the single DSM that we used in Köper and Schulte im Walde (2017b). The textual only performance of this model was .22; the multi-modal performance was .21. The rating filter obtained a significant[5] better score of .27 and the cluster filter increased performance to .25.

Figures 4.12 zooms into target subsets by dividing the GhostPV dataset according to ambiguity (one sense vs. multiple senses), frequency, abstractness vs. concreteness, imageability, and compositionality. The bars refer to the textual model, the multi-modal model (including all images for all targets), and the results obtained when using the dispersion/rating/clustering filters. Zooming into target subsets, the predictions for monosemous targets are better than those for ambiguous targets; the predictions are ditto for low-frequency vs. high-frequency targets. Taking frequency as an indicator of ambiguity, these differences are presumably due to the difficulty of distinguishing between multiple senses in vector spaces that subsume the features of all word senses within one vector, which applies to our textual and multi-modal models. Interestingly, the compositionality of abstract and less imaginable targets is predicted better than for concrete and imaginable targets. However, we clearly notice a large improvement on the concrete subset, when applying the

[5] *Steiger*'s test ($p < 0.001$) (Steiger, 1980)

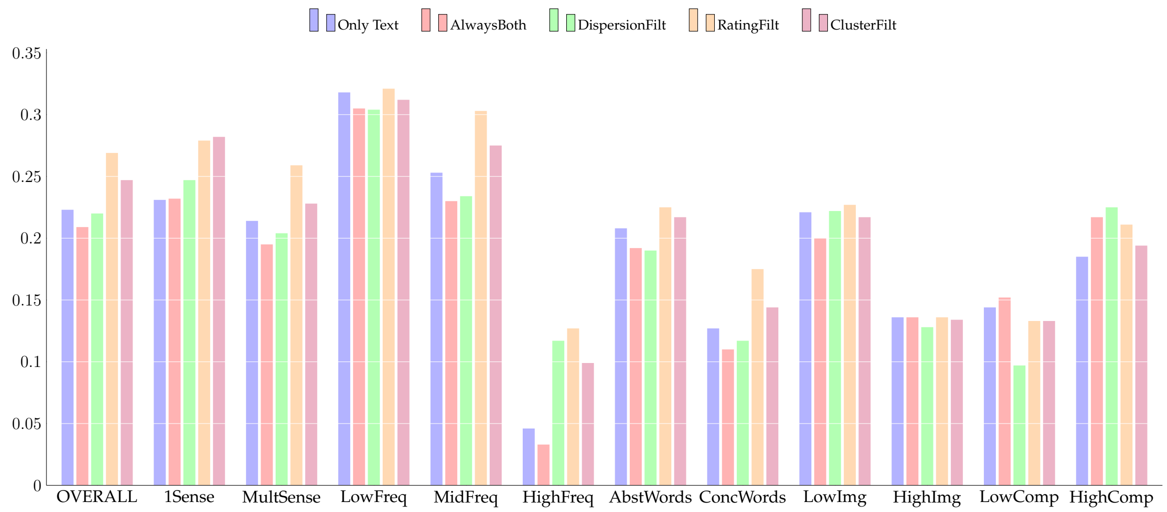


Figure 4.12: Performance on target subsets: GhostPV

rating filter.

Imageability Threshold: So far, our models always defined the threshold for the dispersion and rating filters as the median of a given dataset. Thus half of the data, according to the respective filter, was processed in a multi-modal way and the other half of was treated as non-visual and, therefore, represented by textual modality only. Therefore, Figure 4.13 compares a successive increase of images for the multi-modal model in comparison to the textual model, based on the dispersion and rating filters and GhostPV. The x-axis starts with an only textual modality (left) and reaches a point where both modalities are always used (100% T+V). It can be seen that the multi-modality improves the textual modality in most of the proportions, reaching its maximum when adding images for ≈80% of the most imaginable verbs; when adding the ≈10% of the least imaginable verbs, the model strongly drops in its performance. For the dispersion filter, the tendencies are less clear. We conclude that the visual information adds to the textual information by adding a selection of images (unless they are overly dissimilar to each other, or for non-imaginable targets), because the textual information by itself is poor.

Comparing Noun Compounds: German noun-noun compounds represent another interesting type of MWEs. Here, both constituents are nouns e.g., *Feuerwerk* "fire works" is composed of the nominal constituents "Feuer" *fire* and "Werk" *opus*. In our paper (Köper and Schulte im Walde, 2017b), we systematically compared the

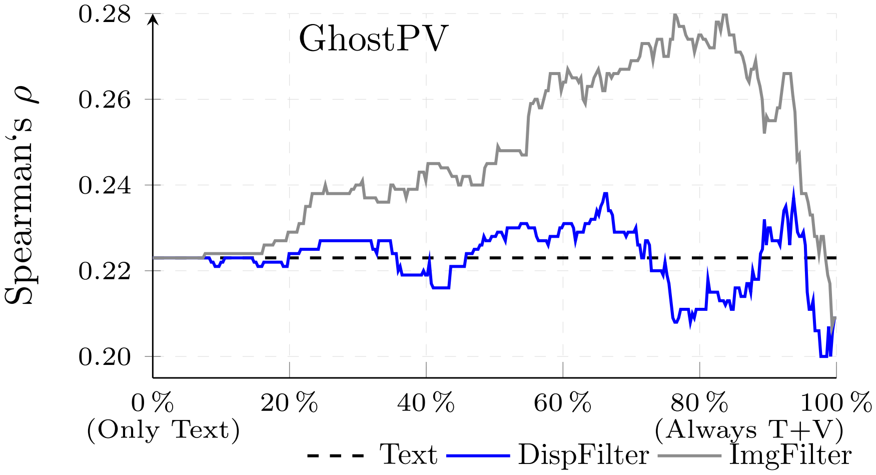


Figure 4.13: Prediction of compositionality (GhostPV): Effect of threshold for dispersion and rating filters

compositionality prediction of PVs to the prediction of noun compounds. Although noun compounds are not the main focus of this thesis, we want to point out that our experiments demonstrated a strong difference in the effect of adding visual features to predict compositionality.

For nouns, we obtained higher performance using the textual modality; in addition we found that the visual modality adds complementary features by adding all available – and potentially noisy– images. Thus, performance could be increased when always using the visual information without filters. We concluded that filter relying on imageability norms and clustering filter positively affect the verbal but not the nominal perceptual feature spaces.

These differences in the results point to questions that are unresolved across research fields: while humans can easily grasp intuitions about the abstractness, imageability and compositionality of nouns, the categorizations are difficult to define for verbs (Glenberg and Kaschak, 2002; Brysbaert et al., 2014). PVs add to this complexity, especially since compositionality (rating) is typically reduced to the semantic relatedness between the complex verb and the BV, ignoring the particle that, however, contributes a considerable portion of meaning to the complex verb.

4.2.6 Summary

Our work investigated and systematically studied various distributional models to predict compositionality for German PVs. Across the two gold standards we have seen that the reconstruction of separated PVs is particularly necessary to enhance the prediction of compositionality. In the same way, we found that the underlying corpus has a strong impact. Here our findings can be summarized as "the bigger, the better" since the 15× larger DECOW corpus clearly provided better representations than the ones from SDEWAC. Dimensionality, unless being too small, had only a minor impact. We obtained best performance with smaller windows for DECOW and windows of 6-10 for SDEWAC. We showed further that adding visual information to a textual model can improve the prediction of compositionality. Here, we experimented with existing and novel visual filtering techniques. Our experiments suggest that visual information for verbs should only be integrated based on information taken from external imageability norms.

4.3 Compositionality as Vector Prediction

We are now going to perform a different approach to model particle-verb constructions. Instead of relying on distributional similarity only, we aim to model the compositional process itself. We rely on the framework of compositional distributional semantics, as extensively discussed in Baroni et al. (2014a).

In the following section we investigate the usage of methods that model the contribution of the particle and aim to predict the distributional meaning of the PV, given its BV. For comparison we additionally evaluate these models on other affixes across German and English. The work presented here (Köper et al., 2016) was the result of a collaboration with Sebastian Padó and Max Kisselew from project B9 *Distributional Characterization of Derivation* within the collaborative research center SFB-732 (Sonderforschungsbereich).

4.3.1 Introduction

Recent work on compositional distributional semantics has addressed the modeling of word formation. Lazaridou et al. (2013) were the first to apply distributional semantic models (DSMs) to the task of deriving the meaning of morphologically complex words from their parts. They relied on high-dimensional vector representations to model the derived term (e.g., *useful*) as a result of a compositional process that combines the meanings of the base term (e.g., *to use*) and the affix (e.g., *ful*). For evaluation, they compared the predicted vector of the complex word with the original, corpus-based vector via cosine similarity. Hence, such a setup requires no external annotated resource for evaluation.

More recently, Kisselew et al. (2015) put the task of modeling derivation into the perspective of zero-shot-learning: instead of using cosine similarities they predicted the derived term by learning a mapping function between the base term and the derived term. Once the predicted vector was computed, a nearest-neighbor search was applied to validate if the prediction corresponded to the derived term. In zero-shot-learning the task is to predict novel values, i.e., values that were never seen in training. More formally, zero-shot-learning trains a classifier $f : X \rightarrow Y$ that predicts novel values for Y Palatucci et al. (2009). It is often applied across vector spaces, such as different domains, including different languages (Mikolov et al., 2013b) or even different modalities such as language and vision (Lazaridou et al., 2015a).

The experiments by Kisselew et al. (2015) were performed over six derivational patterns for German (cf. Table 4.1), including PVs with two different particle prefixes (*an* and *durch*). Their findings showed that in particular the PVs were difficult to predict. PVs such as *anfangen* (to start) are compositions of a BV such as *fangen* (to catch) and a verb particle such as *an*. Predicting PV meaning is challenging because of the high productivity, ambiguity and meaning shifts of German PVs (explained in Section 2.1).

In this work, we focus on predicting the meanings of German PV derivations. Our models provide two contributions to the research field of predicting derivations: (i) We suggest a novel idea of restricting the available training data, which has a positive impact on the mapping quality. (ii) We integrate a correction method for popular nearest neighbors into our models, the so-called *hubs* (Radovanović et al., 2010), to improve prediction quality.

POS	Affix	Example	Inst.
adj/adj	un-	sagbar - unsagbar	80
adj/adj	anti-	religiös - antireligiös	80
noun/noun	-in	Bäcker - Bäckerin	80
noun/noun	-chen	Schiff - Schiffchen	80
verb/verb	an-	backen - anbacken	80
verb/verb	durch-	sehen - durchsehen	80

Table 4.1: German derivation dataset from Kisselew et al. (2015).

POS	Affix	Example	Inst.
verb/verb	auf-	nehmen - aufnehmen	171
verb/verb	ab-	setzen - absetzen	287
verb/verb	mit-	streiken - mitstreiken	216
verb/verb	ein-	laufen - einlaufen	185
verb/verb	zu-	drücken - zudrücken	50
verb/verb	an-	legen - anlegen	221
verb/verb	aus-	malen - ausmalen	280

Table 4.2: German Particle Verb Derivation Dataset.

4.3.2 Prediction Experiments

As in Kisselew et al. (2015), we treat every derivation type as a specific learning problem: we take a set of word pairs with a particular derivation pattern (e.g., "-in", Bäcker::Bäcker**in**), and divide this set into training and test pairs by performing 10-fold cross-validation. For the test pairs, we predict the vectors of the derived terms (e.g., $\overrightarrow{\textit{Bäckerin}}$). The search space includes all the corpus words across parts-of-speech, except for the base term. The performance is measured in terms of recall-out-of-5 (Section 3.2.2), counting how often the correct derived term is found among the five nearest neighbors of the predicted vector.

Derivation Datasets: We created a new collection of German PV derivations[6] relying on the same resource as Kisselew et al. (2015), the semi-automatic derivational lexicon for German *DErivBase* (Zeller et al., 2013). From DErivBase, we induced all the pairs of base verbs and particle verbs across seven different particles. Non-existing verbs were manually filtered out. In total, our collection contains 1 410 BV–PV combinations across seven particles, cf. Table 4.2.

In addition, we apply our models to two existing collections for derivational patterns, the German dataset from Kisselew et al. (2015), comprising six derivational patterns with 80 instances each (cf. Table 4.1), and the English dataset from Lazaridou et al. (2013), comprising 18 derivational patterns (3 prefixes and 15 suffixes) and 7 449 instances (cf. Table 4.3). This allows us to study the particle verb phenomena in contrast to other derivations.

[6] The dataset is available from `http://www.ims.uni-stuttgart.de/data/pv-deriv-dataset`.

POS	Affix	Example	Inst.
verb/adj	-able	believe - believable	227
noun/adj	-al	doctor - doctoral	295
verb/noun	-er	repeat - repeater	874
noun/adj	-ful	use - useful	103
noun/adj	-ic	algorithm - algorithmic	330
verb/noun	-ion	erupt - eruption	687
noun/noun	-ist	drama - dramatist	294
adj/noun	-ity	accessible - accessibility	422
noun/verb	-ize	cannibal - cannibalize	155
noun/adj	-less	word - wordless	172
adj/adv	-ly	diagonal - diagonally	1,897
verb/noun	-ment	equip - equipment	215
adj/noun	-ness	empty - emptiness	652
noun/adj	-ous	religion - religious	207
noun/adj	-y	sport - sporty	454
adj/adj	in-	dispensable - indispensable	151
verb/verb	re-	write - rewrite	136
adj/adj	un-	familiar - unfamiliar	178

Table 4.3: English Derivation dataset from Lazaridou et al. (2013).

Word Embedding Vectors: We relied on the German and English *COW* web corpora[7] Schäfer and Bildhauer (2012) to obtain vector representations. The corpora contain 20 billion words and 9 billion words, respectively. We parsed the corpora using state-of-the-art pipelines integrating the MarMoT tagger and the MATE parser Müller et al. (2013); Bohnet (2010), and induced window co-occurrences for all corpus lemma–POS pairs and co-occurring nouns, verbs, and adjectives in a 5-lemma window. Then, we created 400-dimensional word representations using the *hyperwords* toolkit Levy et al. (2015), with a context distribution smoothing of 0.75 and positive point-wise mutual information weighting together with singular value decomposition. The resulting vector-space models contain approximately 460 000 lemmas for German and 240 000 lemmas for English.

4.3.3 Prediction Methods

Baseline: A baseline method that simply guesses the derived term has a chance of approximately $\frac{1}{460\,000}$ for German and $\frac{1}{240\,000}$ for English to predict the correct term.

[7] `http://corporafromtheweb.org`

Thus, we apply a more informed baseline, the same as in Kisselew et al. (2015), and predict the derived term at exactly the same position as the base terms vector representation.

Additive Method (AvgAdd): *AvgAdd* is a re-implementation of the best method in Kisselew et al. (2015): For each affix, the method learns a difference vector by computing the dimension-wise differences between the vector representations of base term A and derived term B. Thus, the method learns a centroid $\vec{c}$ for all the relevant training pairs (N) with the same affix:

$$\vec{c} = \frac{1}{N} \sum_{i=0}^{n} (B_i - A_i) \tag{4.2}$$

For each PV test instance with this affix, the learned centroid vector is added dimension-wise to the vector representation of the base term to predict a position for the derived term.

Restricting the Training Space (BestAdd): *Avg-Add* learns a vector representation based on the full available training data for each derivational pattern. In this work, we suggest a method *BestAdd$_k$* that restricts the training items of a given base term to those BV–PV training instances that include the k nearest BVs (using $k = 1, 3, 5$) according to their *cosine*. The motivation for our adjusted method is based on the observation that particles are very ambiguous and, thus, differ in their meanings across PVs. For example, the meanings of "an" include a directed contact as in *sprechen::ansprechen* (to speak/to speak to s.o.) and in *schreiben::anschreiben* (to write/to write to s.o.), and also a start of an action as in *spielen::anspielen* (to play/to start playing) and in *stimmen::anstimmen* (to pitch/to start singing). We assume that BVs that are distributionally similar also behave in a similar way when combined with a specific particle, and that a more restricted training set that is, however, specified for BV semantics outperforms a larger training set across wider BV meanings.

LexFun: Another compositional model, is the lexical function model (LexFun). This method aims at learning a matrix that can be used to perform the mapping between a source and a target vector. Assume, we have a list of n paired words $\{x_i, y_i\}_{i=1}^{n}$, in our application a pair is vector of a base term and the derived (target).

Next, a matrix is computed via a least-squares regression:

$$\min_{W} \sum_{i=1}^{n} \|y_i - Wx_i\|^2 \tag{4.3}$$

Using this method, we can now convert any new (unseen) vector from one space (base term) to its derived term. After training (obtaining W), any base word vector x_e can be mapped to the particle verb target by calculating $y_e = Wx_e$.

We also conducted preliminary experiments where we used this method on a smaller, restricted training space (in a similar fashion to *bestadd*$_x$); however, *LexFun* performs clearly better when more training instances are available.

3CosMul: We also re-implemented *3CosMul* (Levy and Goldberg, 2014a), a method that has been proven successful in solving analogy tasks, such as *man* (A) is to *king* (B) as *woman* (C) is to *queen* (D). *3CosMul* does not explicitly predict a position in space but selects a target D in space that is close to B and C but not close to A. We applied *3CosMul* by always using the most similar training instance (as for *BestAdd* with $k = 1$).

Local Scaling: All the methods introduced in the previous section perform a nearest-neighbor search at the predicted position. We suggest to improve the prediction quality at this stage by mitigating the hubness problem (Dinu et al., 2015). *Hubs* are objects in vector space that are likely to appear disproportionately often among nearest neighbors, without necessarily being semantically related. Hubness has been shown to be an intrinsic problem of high-dimensional spaces (Tomasev, 2014). To reduce hubness, three unsupervised methods to re-scale the high-dimensional distances have been proposed (Schnitzer et al., 2014): local scaling, global scaling, and shared nearest neighbors. Dinu et al. (2015) additionally showed some global correction measures to correct for hubness in zero-shot learning; however, their method requires additional source side elements that are not always available. Hubness-aware distance methods aim primarily at repairing asymmetric nearest-neighbor relations, i.e., element a is a nearest neighbor of b but not vice versa. We focus on a local scaling (LS) type of hubness-correcting distance measure, namely the non-iterative contextual measure NI (Jégou et al., 2007):

$$NI(x,y) = \frac{d_{xy}}{\sqrt{\mu_x \cdot \mu_y}} \tag{4.4}$$

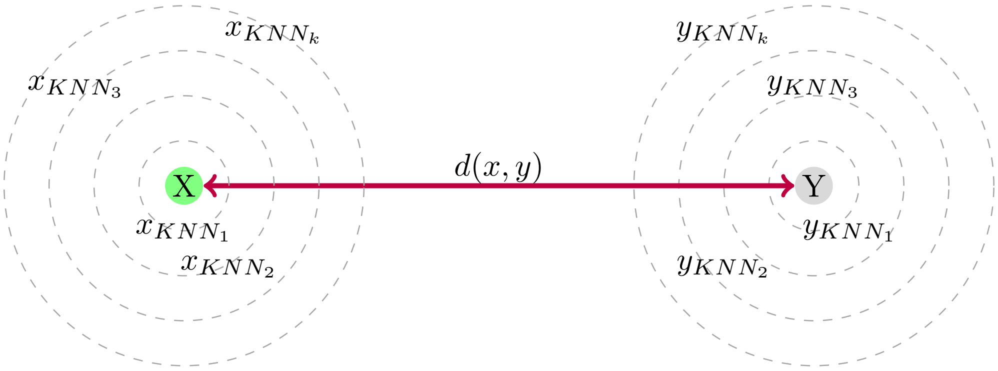


Figure 4.14: Example Illustration of NI (Local-Scaling)

NI relies on the average distance μ of x and y to their k nearest neighbors, as Figure 4.14 illustrates. It increases the similarity between x and y in cases where we observe low average similarities between x, y and its k nearest neighbors. Intuitively, if a word x is not even close to its nearest neighbors but comparably close to y, then, we increase the similarity between x and y. We conducted several preliminary experiments using $k = 3, 5, 10$, and 15; we obtained very little differences with regard to the value of k. In this work we report results for $k = 15$. since it showed a small tendency to perform best.

All of our methods perform local scaling based on a new predicted vector. However, the *3CosMul* method does technically only perform a ranking and does not create a prediction vector; hence, we adapt local scaling. For *3CosMul*, we perform local scaling by scaling over the neighborhood information for all the four parts (A, B, C and D) in the analogy:

$$\textit{3CosMul+LS (D)} = \frac{\textit{3CosMul(D)}}{\sqrt[4]{\mu_A \cdot \mu_B \cdot \mu_C \cdot \mu_D}} \tag{4.5}$$

4.3.4 Results

***BestAdd* and Local Scaling:** Table 4.4 presents macro-averaged recall-out-of-5 scores, giving equal weight to each derivation regardless of the number of instances. Across the three datasets, the default results (i.e., without local scaling) obtained with our novel method *BestAdd* (with $k = \{3, 5\}$) are significantly[8] above *AvgAdd* ($p < 0.01$), the previously best method for the existing German and English datasets. Additionally, we confirm the findings from Kisselew et al. (2015); Melymuka et al. (2017),

[8] Significance relies on χ^2.

Dataset: Method:	**Particle Verbs (DE)** Default	 + NI_{15}	**Kisselew (DE)** Default	 +NI_{15}	**Lazaridou (EN)** Default	 + NI_{15}
Baseline	10.79%		16.08%		15.36%	
AvgAdd	11.82%	+1.28%	24.26%	+3.14%	24.19%	+2.95%
BestAdd_1	10.22%	+1.19%	33.91%	+3.97%	27.32%	+1.87%
BestAdd_3	**14.26%**	**+2.24%**	**38.50%**	**+4.17%**	37.06%	+1.40%
BestAdd_5	14.44%	+1.97%	38.07%	+4.61%	**38.49%**	**+2.12%**
LexFun	3.70%	+1.10%	18.83%	+1.48%	16.69%	-0.40%
3CosMul	10.06%	-0.73%	33.91%	+ 1.04%	27.88%	+0.90%

Table 4.4: Macro-averaged recall-out-of-5 across methods, with and without local scaling NI_{15}.

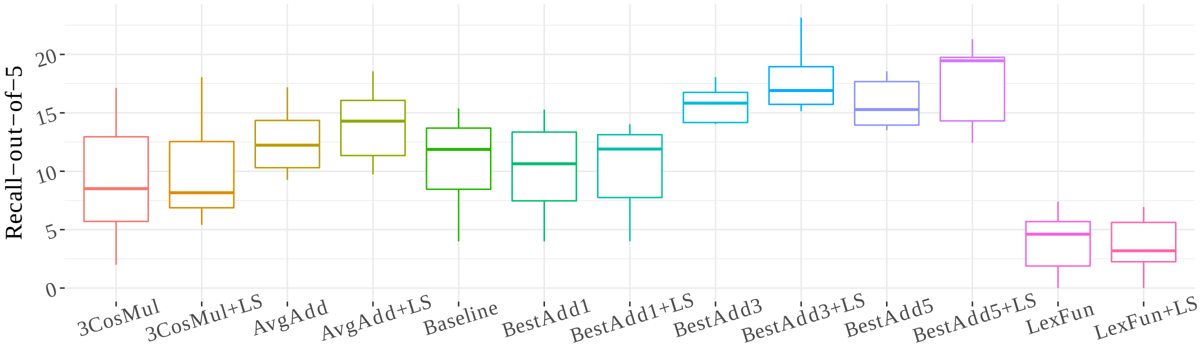


Figure 4.15: Recall-out-of-5 results across methods. Distribution for the German PV derivation dataset.

reporting an inferior performance of *LexFun* in general. *BestAdd* with $k = 1$ and *3CosMul* perform at a similar level than *AvgAdd*, but for our new PV derivation dataset do not even outperform the baseline. Therefore, restricting the training process to a small selection of nearest neighbors has a positive impact on the prediction quality.

Furthermore, local scaling relying on $k = 15$ nearest neighbors (NI_{15}) improves the prediction results in all but one case. However, these improvements are not significant.

A manual inspection of the PV dataset revealed that particularly the baseline, namely the nearest neighbors of the base term, tends to list many neighbors with different part of speech in particular nouns. On the other hand, all prediction methods seem to correctly predict a position that is close to verbs. Although we consider the entire word space, the top predictions contain verbs almost exclusively. With respect to NI_{15}, it is often the case that there are no differences; on the other hand, there are some cases where it enhances the prediction a lot. For example, "aufrappeln" *(to pick oneself up)* is for most methods ≈ the 10th nearest neighbors without NI, using the correction method it is either correctly predicted or the 2nd nearest neighbor to the prediction.

The results in Table 4.4 also demonstrate that predicting PVs is the most challenging derivation task, as the results are significantly lower than for the other two datasets. Figure 4.15 once more illustrates the recall-out-of-5 results for our new PV dataset; it can be seen that the performance across methods exhibits little difference (variance) with respect to performance. Therefore, we conclude that our findings apply across all the particle types. In the following, we zoom into dataset derivation types.

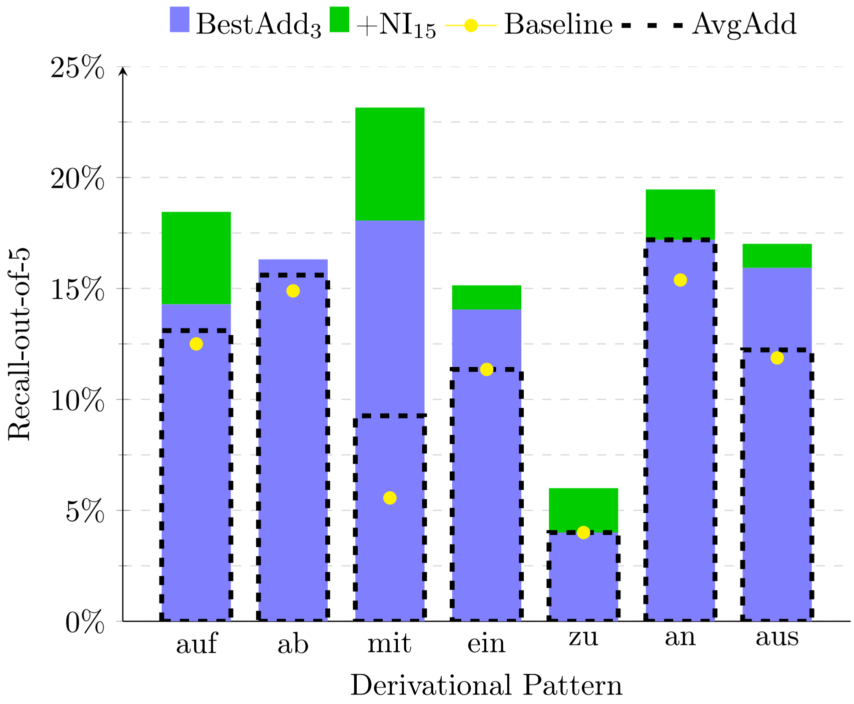


Figure 4.16: Performance gain across particle types.

Improvement across Derivation Types and Languages: Figures 4.16 to 4.18 break down the results from Table 4.4 across the German and English derivation types.

The blue bars show the *BestAdd*3 results, and the green stacked bars represent the additional gain using local scaling (NI$_{15}$). The yellow points correspond to baseline performance, and the dotted black lines correspond to the *AvgAdd* results.

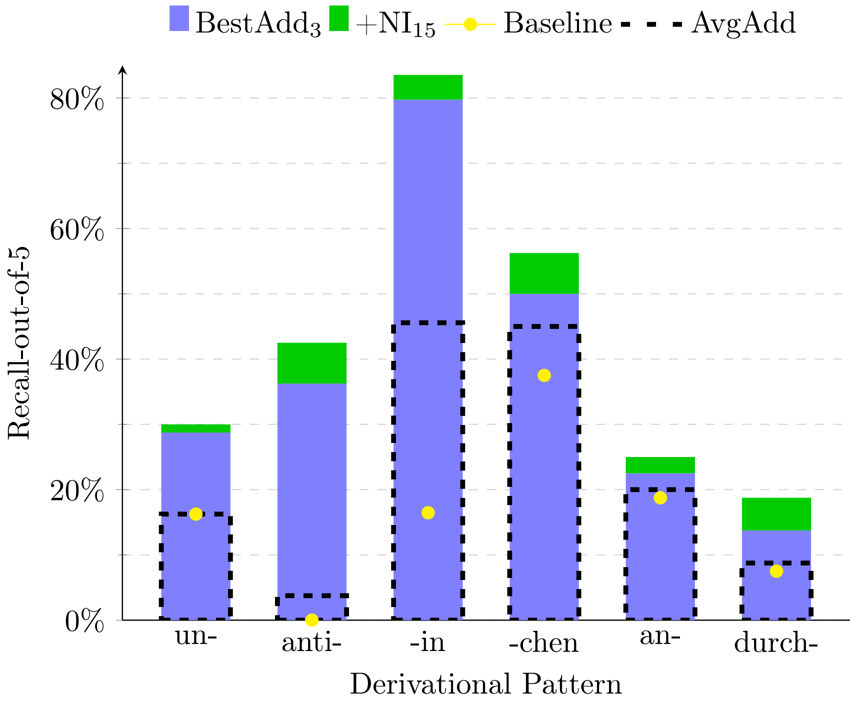


Figure 4.17: Performance gain for derivation types from Kisselew et al. (2015).

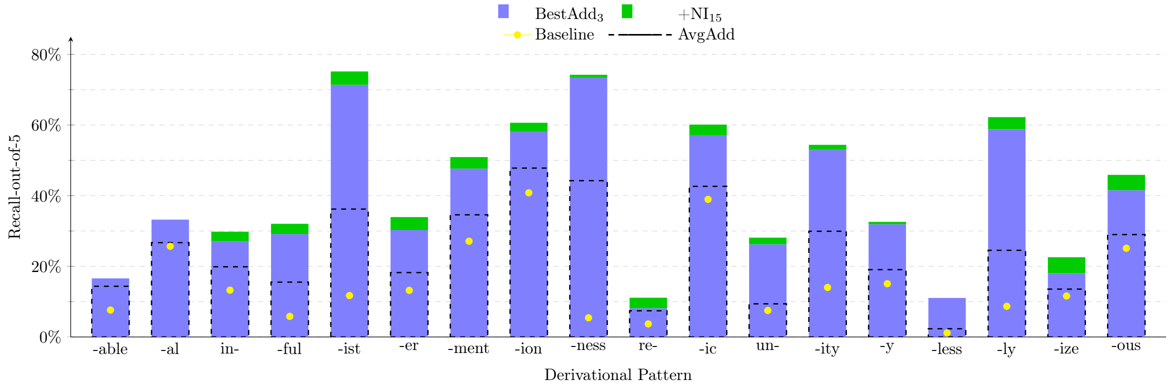


Figure 4.18: Performance gain for derivation types from Lazaridou et al. (2013).

We can see that *BestAdd*3 not only outperforms the previously best method *AvgAdd* on average but also for each derivation type. In addition, local scaling provides an additional positive impact for all but one particle type in German, *ab-*, and for all but three derivation types in English, *-able, -al, -less*.

At the same time, we can see that the impact of local scaling is different across the derivation types. For example, looking into the data, we observe that *mit* PVs are often wrongly mapped to nouns, and *BestAdd* and local scaling correct this behavior: The nearest neighbors of the verb "erledigen" *(to manage sth.)* with *BestAdd*3 are "Botengang" *(errand)*, "Haushaltsarbeit" *(domestic work)*,'Hausmeisterarbeit" *(janitor work)*, and another six compounds with the nominal head "Arbeit" *(work)*. Additional local scaling predicts the correct PV "miterledigen" *(to manage sth. in addition)* as the second nearest neighbor.

Recall-out-of-X Across Particle Types: Figure 4.19 focuses on the particle types, but varies the strength of the evaluation measure. Relying on *BestAdd*3 with local scaling NI_{15}, we apply recall-out-of-x with $x \in [1, 10]$. With one exception (*zu*), all the particle types achieve a performance of 15-23% for recall-out-of-5, so *zu* had a negative impact on the average score in Table 4.4. Looking at recall-out-of-10, the performances go up to 20-30%. PVs with the rather non-ambiguous "mit" are again modeled best, while PVs with strongly ambiguous particles (such as 'an' and "auf") are also modeled well.

4.3.5 Summary

We compared a variety of composition methods to predict the distributional meaning of derivationally related words. We suggested two ways to improve the prediction of derived terms for English and German. We found that (i) particle-verb motivated training-space restrictions and (ii) local scaling to address hubness in high-dimensional spaces had a positive impact on the prediction quality of derived terms across datasets. Particle-specific explorations demonstrated the difficulty of this derivation, and the differences across the particle types.

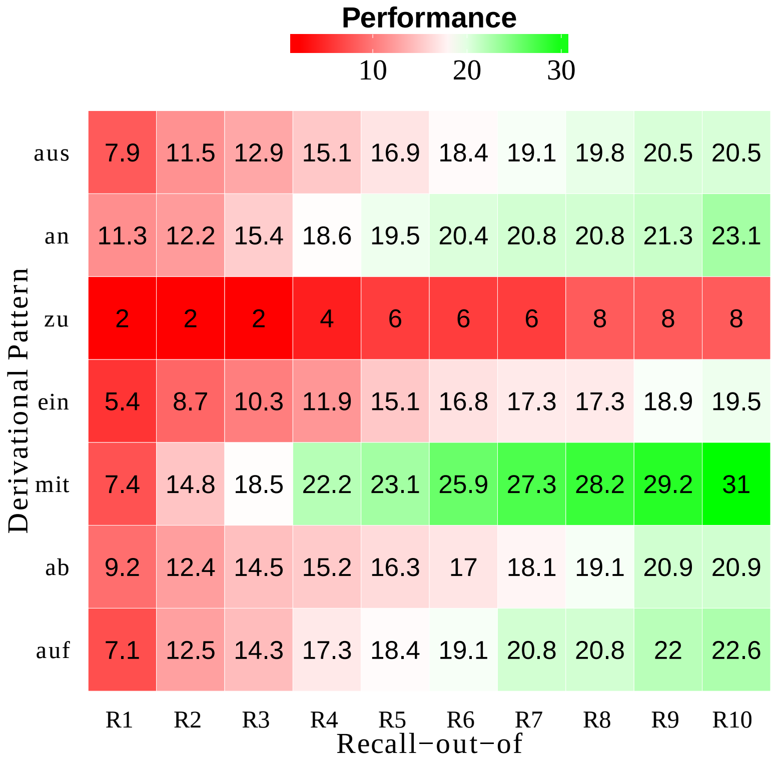


Figure 4.19: Recall-out-of-[1,10] across particles.

4.4 Chapter Summary

This chapter presented experiments to predict particle-verb compositionality judgments and experiments to build PV representations relying on compositional methods.

We have seen that PVs are difficult. Even our best model obtains comparably low correlation with human compositionality judgments. Similar findings were observed in our vector prediction setup; here, PVs obtained the lowest scores in comparison to other derivational patterns.

Furthermore, we have seen that PVs representations are most useful for compositionality prediction when using large amounts of textual data and small window sizes. Most important, our findings suggest that it is necessary to use representations that integrate contextual information from syntactically separated PVs. In the same way, we were able to enhance representations by providing additional visual information combined with clustering and/or imageability norms.

With respect to the compositional experiment, our findings show that we are able to approximate the contribution of the particle best, when learning a vector-transformation only based on similar BVs (training space restrictions). In addition, we propose a better nearest-neighbor search, that accounts for popular neighbors.

5 Sense Discrimination

The previous chapter focused on the modeling of compositionality for German PVs. This was done by applying a type-based perspective, that is without looking at individual contexts. In the previous experiments, we approximated the meaning of a verb across all contexts and stored it in a single vector representation.

In this chapter, we study the meaning of PVs in context. In short, we explore computational models to detect different meanings of PVs and to represent distinct meanings of a particle verb. First, we present a binary distinction between literal and non-literal usage in Section 5.1. Here the task is to decide if the given PV in context is used metaphorically or not. Our goal is to model and represent fine-grained senses of PVs in Section 5.2. Here, we explore methods to learn multi-sense representations, as well as different applications for such representations.

The experiments described in this chapter are published in Köper and Schulte im Walde (2017b) and Köper and Schulte im Walde (2017a).

5.1 Token-based Non-Literal Language

Automatic detection of non-literal expressions (including metaphors and idioms, see Section 2.3) is critical for many natural language processing tasks such as information extraction, machine translation, and sentiment analysis. For this reason, the last decade has seen an increase in research on identifying literal vs. non-literal meaning (Birke and Sarkar, 2006, 2007; Li and Sporleder, 2009; Sporleder and Li, 2009; Turney et al., 2011; Shutova et al., 2013; Tsvetkov et al., 2014), as well as the establishment of workshops on metaphorical language in NLP.[1]

In this section, we explore the prediction of literal vs. non-literal language usage of a computationally challenging class of multiword expressions: German PVs. As explained in Section 2.1, PVs are highly productive, and the particles are notoriously ambiguous. Furthermore, the particles often trigger (regular) meaning shifts when

[1] `sites.google.com/site/metaphorinnlp2016/`

they combine with BVs; therefore the resulting PVs represent frequent cases of non-literal meaning.

The following experiments present a binary, token-based classification of German PVs into literal vs. non-literal usage. We create a novel and large collection of PVs and context, annotated according to their literal or non-literal usage. A random forest improving standard features (e.g., bag-of-words; affective ratings) with PV-specific information and abstraction over common nouns significantly outperforms the majority baseline. In addition, we carry out PV-specific classification experiments. These experiments demonstrate the role of shared particle semantics and semantically related BVs in PV meaning shifts.

In the following sections, we describe previous work on non-literal language identification and metaphor detection systems (Section 5.1.1), before we introduce our dataset on German PVs (Section 5.1.2), the particle-verb features (Section 5.1.3), and the experiments, results and analyses (Section 5.1.4).

5.1.1 Related work

Previous work relevant to our work includes research on identifying non-literal language usage, metaphor detection and computational work on (German) PVs (already mentioned in Section 2.1.2).

Birke and Sarkar (2006), Birke and Sarkar (2007), Li and Sporleder (2009) and Sporleder and Li (2009) performed binary token-based classifications for English datasets, relying on various contextual indicators. Birke & Sarkar exploited seed sets of literal vs. non-literal sentences, and used distributional similarity to classify English verbs. Li & Sporleder defined two models of text cohesion (a cohesion chain and a cohesion graph) to classify V+NP and V+PP combinations. Shutova et al. (2013) performed both metaphor identification and interpretion (by paraphrasing), focusing on English verbs. She relied on a seed set of annotated metaphors and standard verb and noun clustering, to classify literal vs. metaphorical verb senses. Gedigian et al. (2006) also predicted metaphorical meanings of English verb tokens, however, heavily relying on manual rather than unsupervised data (i.e., labeled sentences and PropBank annotation) and a maximum entropy classifier. Turney et al. (2011) assumed that metaphorical word usage is correlated with the degree of abstractness of the word's context, and classified word senses in a given context as either literal or metaphorical. Their targets were adjective–noun combinations and verbs. Tsvetkov et al. (2014) presented a language-independent approach to metaphor identification. They used affective ratings, WordNet categories, and vector-space word representations to train a metaphor-detecting classifier on English samples and, then, applied it to a different target language using bilingual dictionaries. Recently, Shutova et al. (2016) showed successfully that visual information taken from a search engine can be used to detect metaphorical usage for phrases. Other commonly seen features used for metaphor detection are selectional preferences (Martin, 1996; Shutova and Teufel, 2010; Shutova et al., 2010; Haagsma and Bjerva, 2016), abstractness (see Section 3.3), topic models (Heintz et al., 2013) and word embeddings (Dinh and Gurevych, 2016).

5.1.2 Particle-Verb Dataset

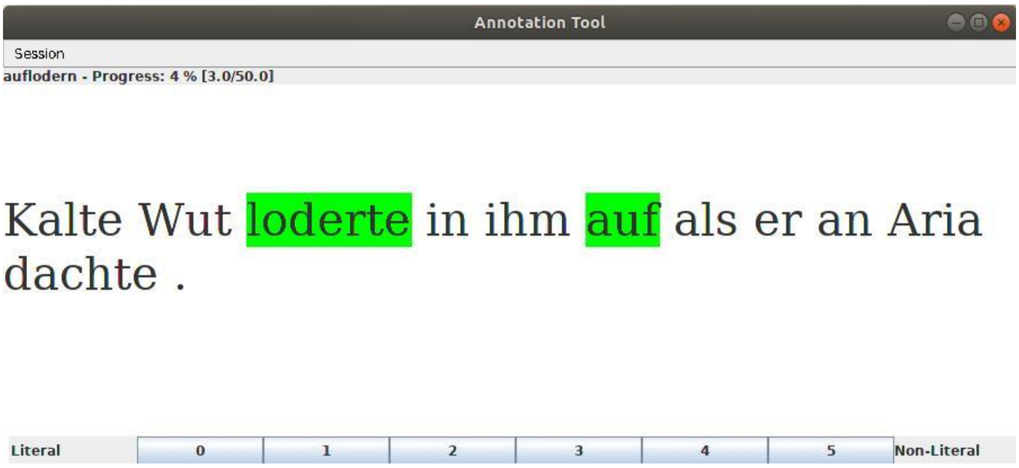
We selected 165 PVs across 10 particles, based on previous experiments and datasets that incorporated German PVs with regular meaning shifts, various degrees of ambiguity, and across frequency ranges (Springorum et al., 2013b,a; Bott and Schulte im Walde, 2015). For the 165 PVs, we randomly extracted 50 sentences from *DECOW14AX* (Section 3.5). Combining part-of-speech and dependency information, we were able to reliably sample both separated and non-separated PV occurrences ("Der␣Ast␣**bricht␣ab**" vs. "Der␣Ast␣ist␣**abgebrochen**").

Three German native speakers with a linguistic background annotated each of the 8 128 sentences[2] on a 6-point scale $[0, 5]$, ranging from clearly literal (0) to clearly non-literal (5) usage. We wrote a simple annotation tool for this purpose. Figure 5.1 shows a screenshot of the tool. It can be seen that the tool highlights the target verb and the particle. Although we provided a scale instead of a binary decision, we noticed that all the annotators showed a tendency to use the extreme classes (0 or 5) most frequently. Figure fig:hist illustrates this behavior for each annotator.

The total agreement of the annotators on all six categories was 43%, obtaining a fair Fleiss' $\kappa = 0.35$. Dividing the scale into two disjunctive ranges with three categories each ($[0, 2]$ and $[3, 5]$), the total agreement of the annotators on the two categories was 79% with a substantial Fleiss' $\kappa = 0.70$. In the following experiments, we used the binary-class distinction, and we disregarded all cases of disagreement. Starting from 8 128 sentences, we removed 1 692 (20.8%) sentences. This final dataset comprises 6 436 sentences: 4 174 literal and 2 262 non-literal uses across 159 PVs and 10 particles.[3] Figure 5.3 shows the distribution of literal and non-literal sentences

[2] Some PVs appeared < 50 times in the corpus.

[3] The dataset is accessible from `http://www.ims.uni-stuttgart.de/data/pv_nonlit`.



auflodern - Progress: 4 % [3.0/50.0]

Kalte Wut loderte in ihm auf als er an Aria dachte .

Figure 5.1: Screenshot of the annotation tool and procedure.

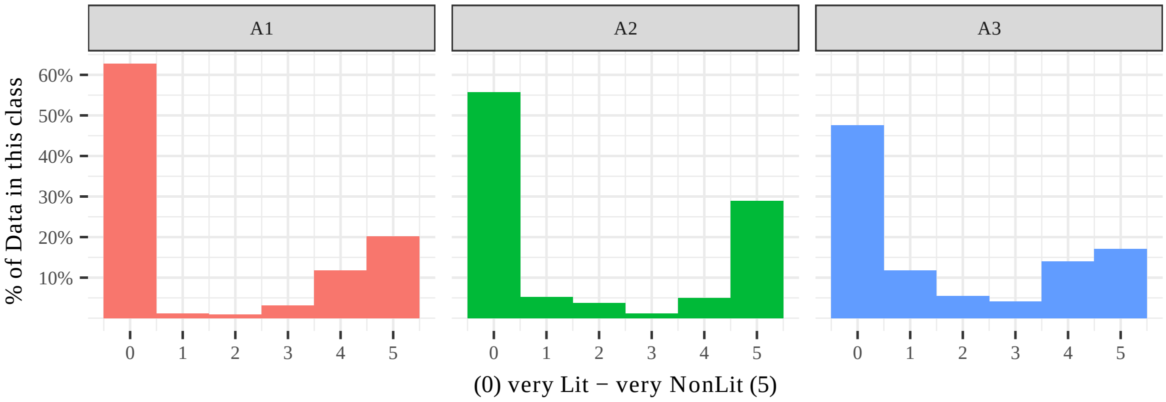


Figure 5.2: Histogramm: How the Annotators applied the 6-point scale $[0, 5]$.

across the particles. It can be seen that across particle, we find a high amount of non-literal language usage. Note that "durch" is an outlier, as it includes only three highly non-literal PVs.

5.1.3 Features

Our feature space includes standard features to detect non-literal language uses (bags-of-words and affective ratings) as well as PV-specific features and abstraction over common nouns.

Unigrams: As a standard feature in vector-space models, we used all the words in the particle verb sentences, i.e., a bag-of-words model relying on unigrams. We expected this standard information to be useful, because some words such as the

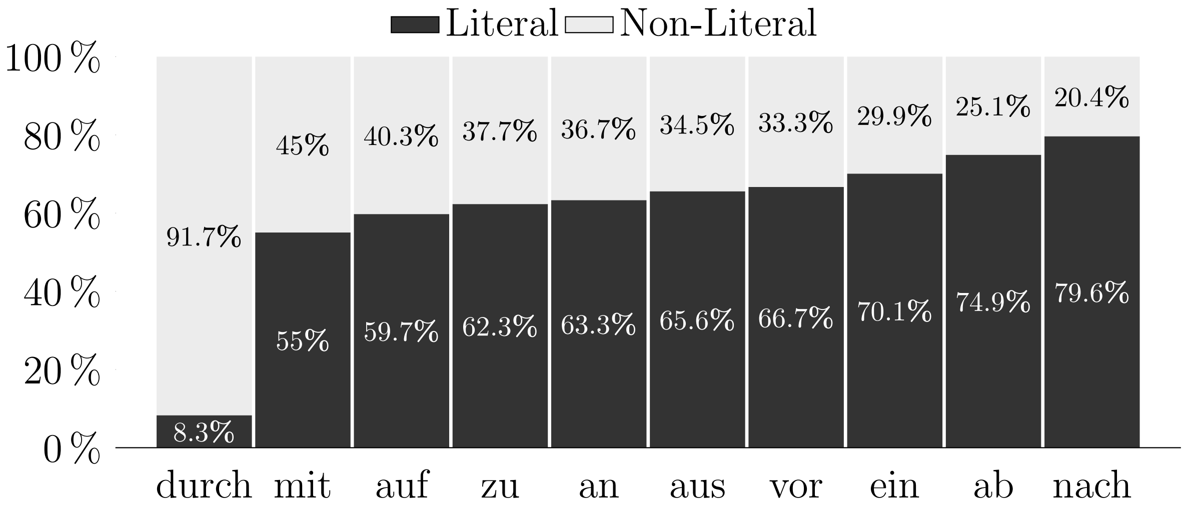


Figure 5.3: Lit/Non-lit distribution across particles.

abstract noun "Hoffnung" (*hope*) and the concrete noun "Geld" (*money*) frequently occur with non-literal rather than with literal language usage:

1. (non-lit.) "Die Hoffnung **keimte** früh **auf**."
   *That hope **arose (lit: sprouted)** early.*

2. (non-lit.) "Es sollen keine Hoffnungen **aufgetischt** werden."
   *No hope should be **provided (lit: dished up)**.*

3. (non-lit.) "Er versucht das Geld **abzugraben**."
   *He tries to **demand (lit: dig off)** the money.*

4. (non-lit.) "Du kannst dir nicht selbst Geld **zuschaufeln**."
   *You cannot **lit:shovel up** money to yourself.*

Therefore, we count how many times we have seen a word in the literal and in the non-literal example. Although we have comparably small amount of training data, it is possible that some words repeat within literal or non-literal language usage. This information is especially useful for the detection of idiomatic expressions (see Section 2.3.2), as they appear always with the same context patterns.

To overcome data sparseness, we did not use the unigrams as individual features ($|V|$ = feature space), but implemented this feature as the output of a text-classifier. We relied on the Multinomial Naive Bayes (MNB, explained in Section 3.2) classifier by McCallum and Nigam (1998). While the classifier was designed for document classification, we considered a sentence as a document and the possible class outcomes were literal and non-literal.

Noun Clusters: Because of the severe data sparseness in our PV feature sets, we performed noun generalization and applied the generalized information to all nouns in our PV contexts. Using all the approximately 430 000 nouns that appeared >100 times in the *DECOW14AX* corpus, we applied k-Means clustering with $k \in [2, 10\,000]$ based on a distributional representation of the nouns. As an alternative to the standard unigrams, we then replaced every noun in the PV sentences with its corresponding cluster tag. For example, each occurrence of the noun Hoffnung *(hope)* would be replaced with an artificial symbol representing its cluster: Hoffnung→ NN_{42}.

Affective Ratings: Previous work on detecting non-literal language often makes use of psycholinguistic attributes, namely abstractness and concreteness ratings (Turney et al., 2011), and imageability ratings (Tsvetkov et al., 2014). Words with high abstractness ratings refer to entities that cannot be perceived with our senses; a large subset of which are non-visual (i.e., receive low imageability). It has been shown that non-literal expressions tend to occur with abstract words (*dark humor* versus *dark hair*). Thus, we expected affective ratings to be useful for PVs as well. Looking at the actual data, reveals that this information is indeed useful. Table 5.1 lists the ten most helpful unigram features for each class. The counts were obtained on the entire data. Freq refers to the total frequency in our collection (literal and nonliteral usage). It can be seen that the top features for the literal class are highly concrete (wall, moon, whole, tree, color,skin)[4]. On the other hand, we see very abstract concepts, unigrams for the abstract class include hope, information, love, quiet, conflict, and discussion.

Hence, we reimplemented the algorithm from (Turney and Littman, 2003) to create large-scale abstractness and imageability ratings for German (Köper and Schulte im Walde, 2016). This process was explained in more detail in Section 3.4.1. Based on these ratings, we defined the following (partially redundant) features for the PV sentential contexts:

1. Rating of the PV subject

[4] The high occurrence of left/right is due to the verb *abbiegen (to turn)*.

Top NonLit			**Top Lit**		
Word	**%NonLit**	**Freq**	**Word**	%NonLit	**Freq**
Hoffnung *(hope)*	97.44%	39	rechts *(right)*	2.17%	46
Geld *(money)*	93.33%	15	links *(left)*	2.38%	42
Information (information)	93.33%	15	Mond *(moon)*	3.45%	29
Liebe *(love)*	92.31%	13	Saft *(juice)*	3.57%	28
Ruhe *(quiet)*	91.67%	12	Loch *(whole)*	3.57%	28
Konflikt *(conflict)*	91.67%	12	Baum (tree)	3.85%	26
Leben *(live)*	91.67%	12	Holz *(wood)*	4.17%	24
Streit *(dispute)*	90.91%	11	Wand *(wall)*	4.35%	23
Dampf *(steam)*	90.91%	11	Farbe *(color)*	4.35%	23
Diskussion *(discussion)*	90.91%	11	Haut *(skin)*	4.55%	22

Table 5.1: Top 10 words for Literal and respectively Non-Literal class.

2. Rating of the PV object

3. Average rating of all nouns (excluding proper names)

4. Average rating of all proper names

5. Average rating of all verbs, excluding the PV

6. Average rating of all adjectives

7. Average rating of all adverbs

While features 3–7 have been adopted from Turney et al. (2011), features 1–2 represent additional, PV-specific features.

Distributional Fit of PV, BV and Context: Particle verbs with a meaning shift are non-compositional regarding their BVs. Thus, we implemented a PV-specific feature that measures the distributional fit of PVs and their BVs in the PV contexts. For example, looking at the following two PV sentences containing the BV *klingen* (to sound), the context of the first, literal sentence fits well to the BV meaning, but the context of the second, non-literal sentence does not. Therefore, the distributional fit of the BV in the literal context should be high, but the distributional fit of the BV in the non-literal context should be low.

1. (lit.) "Der Ton der Gitarre **klingt aus**."
   *The tone of the guitar **fades**.*

2. (non-lit.) “Den Abend lassen wir mit Wein **ausklingen**.”
*We **end** the evening with wine.*

3. (lit.) “In der Diskothek **tanzte** er die Frau **an**.”
In the disco, he dance toward the woman.

4. (non-lit.) “Mein Chef lies mich früh morgens im Büro **antanzen**.”
My boss forced me to show up (lit: dance) early in the morning at the office.

To measure the distributional fit of PVs and BVs to PV contexts, we created 400-dimensional word representations using the *hyperwords* toolkit (Levy et al., 2015) and the *DECOW14AX* corpus. We relied on a symmetrical window of size 3 and applied positive pointwise mutual (PPMI) feature weighting together with SVD. Traditional window-based approaches would mix particle verb and base verb in cases where the particle verb is separated (literal example sentence). Therefore, it is important to apply the particle-verb reconstruction step. This preprocessing includes gluing all separated PVs together, as well as removing words that appeared less than 100 times in the corpus. Based on the word representations, we calculated cosine similarities between the PVs and their contexts, and likewise between the respective BVs and the PV contexts. The contexts we used were the same seven dimensions we used for the affective ratings (cf. Section 5.1.3). For example, regarding the (literal) sentence “Die Katze **springt** auf den Tisch” (*The cat **jumps** on the table*), we calculated the distributional similarity between the PV ”aufspringen” and the subject ”Katze”, and the distributional similarity between the BV ”springen” and the subject ”Katze”, etc. Each PV–context and each BV–context dimension represents an individual feature. We indirectly model the selectional preferences of the verbs, assuming that the preferences of a verb regarding its argument is expressed in terms of their vector similarities.

5.1.4 Classification Experiments

In this section, we present a series of binary classification experiments to distinguish between literal and non-literal PV usage. First, we present the main experiments comparing our features in a global classification setup and, then, we present PV-specific additional experiments that zoom into the role of particle types and into the role of semantically related PVs and BVs. Finally, we provide a qualitative analysis of the features and the errors made by our system.

	Feature Type	$\|f\|$	Lit. F_1	Non-Lit. F_1	*Acc.*	*Acc.* + *P*
1	Majority Baseline	0	78.7	0.0	64.9	-
2	Unigram	12 427	83.2	55.5	75.6	76.5
3	Unigram + NN Clusters	6 305	81.6	66.7	76.3	79.3
4	AC Ratings	7	81.3	60.7	74.7	76.3
5	IMG Ratings	7	77.5	48.1	68.6	71.6
6	Distributional Fit	14	83.0	61.8	76.5	80.2
7	Comb. (2+4+6)	22	88.6	77.1	84.8	86.6
8	Comb. (3+4+6) + NN Clusters	22	88.8	77.3	85.0	86.8

Table 5.2: Results across feature types and their combinations.

Global Classification

We used a random forest (see Section 3.2) with multiple (in our case 100) random trees, with each tree voting for an overall classification result. The algorithm is said to be less prone to overfitting. Besides, a random forest was used successfully for metaphor detection in Tsvetkov et al. (2014). Experiments with other classification methods showed similar but inferior performance. Simple Logistic Regression performed 2nd best. The unigram information was represented by stacking the output of a Multinomial Naive Bayes text classifier as a single feature into the random forest. In addition, we used 10-fold cross-validation. For the machine learning algorithms we relied on the WEKA toolkit (Witten et al., 2011).

The experiments were performed in two modes, (a) without knowledge of the particle (i.e., the individual particle was not provided as a feature), and (b) with explicit knowledge of the particle. In this way, we could identify the contribution of the particle.

The Table 5.2 show the classification results. We report on the feature type, and on the size[5] of the feature set f. We further present literal and non-literal f-scores F_1, and accuracy with and without particle knowledge. We compare against the majority baseline (literal). The columns in Table 5.3 indicate whether the differences in performance are statistically significant, using the χ^2 test and $*$ for $p < 0.001$ and $\circ$ for $p < 0.05$ to mark significance.

The results demonstrate that the classification results across all feature types are significantly better than the majority baseline. The single best performing feature

[5] Remember from Section 5.1.3 that the unigram information is based on all tokens (12 427) but we implemented the unigrams as a single feature (using the output of a classifier), thus the combined setting is only based on 22 features.

	1	2	3	4	5	6	7	8
1								
2	*/*							
3	*/*	-/*						
4	*/*	-/-	○/*					
5	*/*	*/*	*/*	*/*				
6	*/*	*/*	-/-	○/*	*/*			
7	*/*	*/*	*/*	*/*	*/*	*/*		
8	*/*	*/*	*/*	*/*	*/*	*/*	-/-	

Table 5.3: Statistical significance of differences $Acc/Acc + P$ using χ^2.

type (cf. lines 1–6) is the unigram information; in combination with the particle information ($+P$), the distributional PV/BV–context fit is best. Combining the best feature types (2+4+6) once more improves the results, and ditto when adding noun cluster information.[6] We can also see that abstractness (AC) ratings outperform imageability (IMG) ratings.

Therefore, overall, the best performing feature set successfully combines unigrams that incorporate clusters for noun generalization, abstractness ratings, and PV-specific information regarding the distributional PV/BV–context fit and knowledge about the particle. This setup correctly classifies literal sentences with an f-score of 88.8 and non-literal sentences with an f-score of 77.3; overall accuracy is 86.8 over a baseline of 64.9.

It is difficult to compare our results against previous approaches on different datasets and in different languages. Regarding the closest approaches to our work, Tsvetkov et al. (2014) report an accuracy score of 82.0 using 10-fold cross-validation on a training dataset with a majority baseline of 59.2, combining multiple lexical semantic features on a dataset of 1 609 English subject–verb–object triples. Birke and Sarkar (2007) trained a single classifier for each of twenty-five verbs in the English TroFi verb dataset and reported only an average f-score: 64.9 against a majority baseline of 62.9. Turney et al. (2011) obtained an average f-score of 63.9 and additionally report an accuracy score of 73.4 on the same dataset, using abstractness ratings.

In contrast to our work, the two approaches by Birke and Sarkar (2007) and Turney et al. (2011) treated each group of sentences for a given target verb as a separate learning problem, while we learn one classifier across different verbs. Our method

[6] The best cluster analysis in our experiments contained 750 noun clusters.

4 (AC Ratings) can be considered a German re-implementation of the approach by Turney et al. (2011). In comparison to the results of previous work, our approach can safely be considered state-of-the-art.

Noun Cluster Evaluation: As described in Section 5.1.3, we evaluated every possible noun cluster extrinsically. The resulting performance for different k is shown in Figure 5.4. We distinguish between three settings: (i) the performance using all word classes and noun clusters (yellow), the performance using only the noun clusters (blue) (ii), and a setting (iii) where we allowed all word classes, used clusters, and gave the particle as additional information (pink). The dashed line represents the performance when guessing always literal (majority baseline). It can be seen that every $k > 9$ already improves performance over the no-cluster setup. In addition, we see words from all the word classes perform superior to the only noun setting. The best performance is obtained using $k = 750$ which corresponds to ≈573 elements per cluster, if we assume balanced clusters. Furthermore, using all the possible features with noun clusters and the additional particle feature works best.

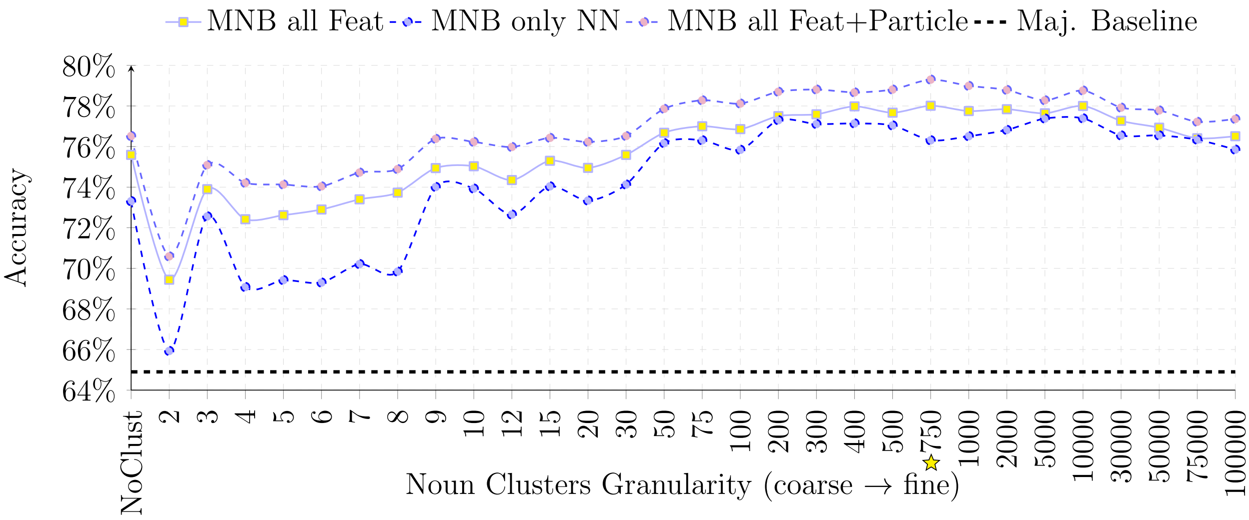


Figure 5.4: Evaluating different Noun clusters granularities

PV Specific Classification

Incorporating Standard Measures of Multiword Idiomaticity: One traditional line of research to identify type-based multiword collocations or idiomatic expressions relies on the association strength between the multiword parts (Evert and Krenn, 2001; Krenn and Evert, 2001; Stevenson et al., 2004): The stronger the association between the parts of a multiword expression (as determined by raw frequency, some variant of mutual information, etc.), the stronger the collocation/idiomaticity of the combination of the parts. Based on this assumption, we calculated the association strength between PVs and their contextual subjects/objects, using *local mutual information (LMI)*, cf. Evert (2005). The LMI scores were based on type-based frequency counts in the *DECOW14AX* corpus and added as features to the respective contexts, assuming that large LMI scores indicate non-literal PV usage.

Adding the LMI values to the overall best feature set from the main experiments decreased accuracy from 86.8 → 86.0. Using the LMI association strength values of the PV–subject and PV–object pairs by themselves provided slightly but non-significantly better results in comparison to the majority baseline: 65.9 $>$ 64.9. We also experimented with the other five contextual feature dimensions; however, the results were even worse. Manual investigations revealed that verb–noun pairs with high LMI scores represent collocations in many cases, but the collocations are not only used in non-literal language but also in literal language, e.g., "Schichten auftragen" *(to apply layers)* or "Sendung ausstrahlen" *(broadcast a program)*.

Domain Adaptation for PVs: Integrating the particle as a feature can be considered as multi-domain adaptation, with a domain represented by a specific particle type. We applied the *Frustratingly Easy Domain Adaptation* method from Daumé III (2007) to PV domain adaptation: While the original (global domain) feature dimension remained the same as before, we expanded the size of our feature space by 10 times, because we added all feature dimensions for all our 10 possible particle types. In other words, in addition to the original feature space, we determined all the features in the feature space for each particle. Unfortunately, this experiment using a random forest classifier provided worse results than before (88.6 → 86.5)

Non-Literality across Particles

To explore the predictability of literal vs. non-literal uses with respect to specific particles, we trained the best classifier from the main experiments on all PVs with particle X and applied the classifier to all PVs with particle Y. Our hypothesis was that pairs of particles with similar ambiguities might predict each other better than pairs with different particle meanings.

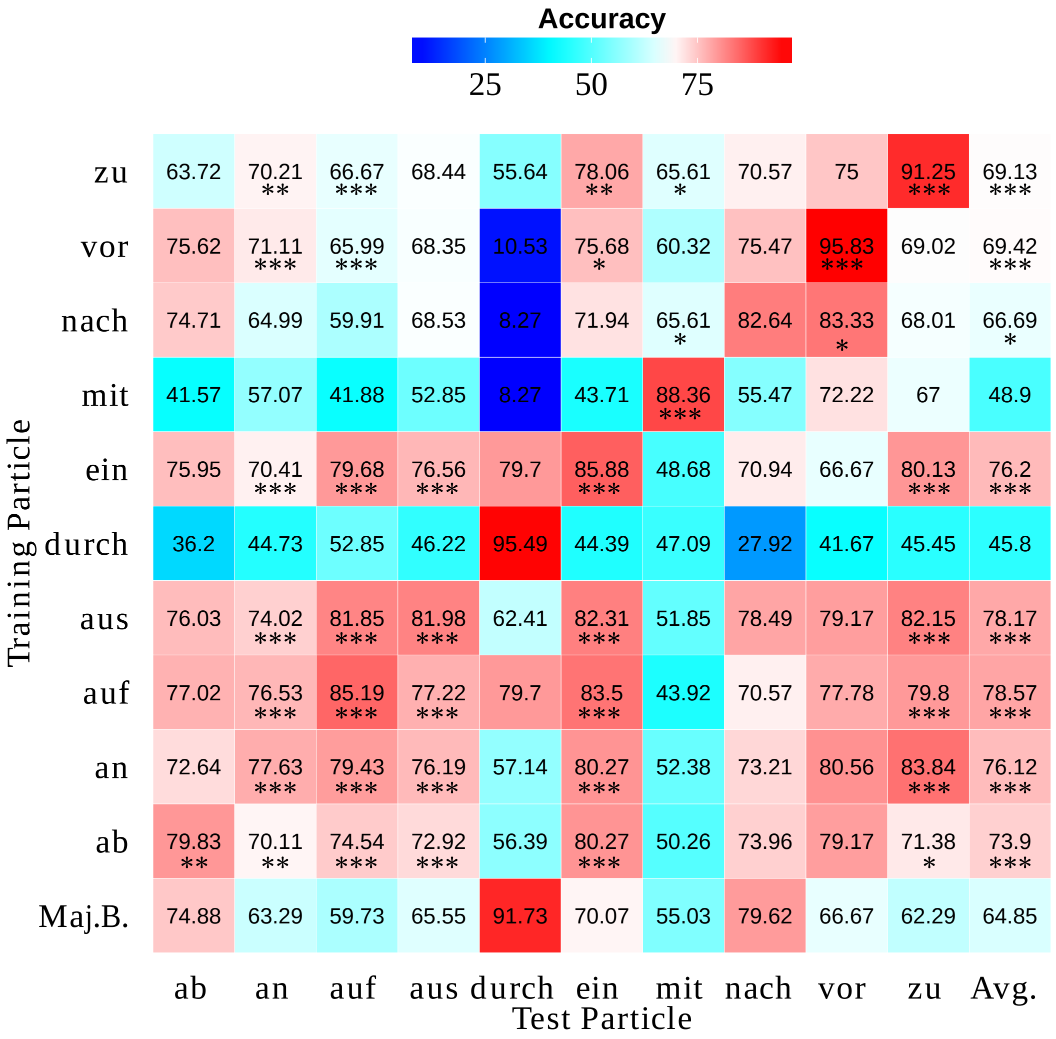
This PV-specific setup could also be applied within a PV group with the same particle: We trained the classifier on all PVs with particle X except for one and, then, we applied the trained classifier to the missing PV with particle X. The setup was repeated for all PVs with particle X, and the average accuracy was calculated. This within-particle setup is similar to cross-validation, where the folds depend on the verbs.

Figure 5.5 provides the results as a heat map, with red indicating high and blue indicating low accuracy scores. The vertical particles on the left correspond to the training particles, and the horizontal particles at the bottom correspond to the test particles. The bottom line shows the majority baseline. For example, training a classifier on "ein" PVs and evaluating it on "aus" PVs results in an accuracy of 76.56, which is significantly better ($* * *$ for $p < 0.001$) than the baseline for "aus" (65.55).

The diagonal in the heat map (showing the within-particle setup) provides particularly high accuracy scores; therefore, the PVs with the same particle predict (non-)literality within the group very well. This demonstrates that the meanings and the meaning shifts across PVs with the same particle (e.g., "aufdecken" and "auftischen") are quite regular. A comparably strong prediction is found between "vor" (*before/in front of*) and "nach" (*after/behind*), with both particles carrying highly similar temporal and local senses. Other examples of strongly related antonymous particle pairs are "auf"/"zu", "ein"/"aus", and "aus"/"an". Examples of strongly related synonymous particle pairs are "an"/"ein", and "aus"/"zu". The particle "durch" correlates poorly with all other particles, which is probably due to the few sentences we collected from the corpus. Similarly, "mit" also correlates poorly with all other particles, because it is the only particle with little ambiguity. Therefore, overall, the heat map corresponds to intuitions about semantic relatedness across particle pairs. Interestingly, many particle pairs show asymmetric results; for example a classifier trained on "an" instances performs well on "zu", while a classifier trained on "zu" shows only mediocre results when validated on "an".

Non-Literality across Particle Verbs

An even more fine-grained experiment setting explored the predictability of a specific particle verb based on the classifier trained on a different particle verb. Our hypothesis was that pairs of PVs that predict each other particularly well share some meaning aspects, either (i) because the training and the test verb share the same BV (**SameBV**: ab*graben*:auf*graben*), or (ii) the PVs are synonymous according to the



Training Particle	ab	an	auf	aus	durch	ein	mit	nach	vor	zu	Avg.
zu	63.72	70.21 **	66.67 ***	68.44	55.64	78.06 **	65.61 *	70.57	75	91.25 ***	69.13 ***
vor	75.62	71.11 ***	65.99 ***	68.35	10.53	75.68 *	60.32	75.47	95.83 ***	69.02	69.42 ***
nach	74.71	64.99	59.91	68.53	8.27	71.94	65.61 *	82.64	83.33 *	68.01	66.69 *
mit	41.57	57.07	41.88	52.85	8.27	43.71	88.36 ***	55.47	72.22	67	48.9
ein	75.95	70.41 ***	79.68 ***	76.56 ***	79.7	85.88 ***	48.68	70.94	66.67	80.13 ***	76.2 ***
durch	36.2	44.73	52.85	46.22	95.49	44.39	47.09	27.92	41.67	45.45	45.8
aus	76.03	74.02 ***	81.85 ***	81.98 ***	62.41	82.31 ***	51.85	78.49	79.17	82.15 ***	78.17 ***
auf	77.02	76.53 ***	85.19 ***	77.22 ***	79.7	83.5 ***	43.92	70.57	77.78	79.8 ***	78.57 ***
an	72.64	77.63 ***	79.43 ***	76.19 ***	57.14	80.27 ***	52.38	73.21	80.56	83.84 ***	76.12 ***
ab	79.83 **	70.11 **	74.54 ***	72.92 ***	56.39	80.27 ***	50.26	73.96	79.17	71.38 *	73.9 ***
Maj.B.	74.88	63.29	59.73	65.55	91.73	70.07	55.03	79.62	66.67	62.29	64.85

χ^2 test : $***$ for $p < 0.001$, and $**$ for $p < 0.01$ and $*$ for $p < 0.05$.

Figure 5.5: Train a classifier on PVs with particle X and test it on PVs with particle Y.

German *Duden*[7] dictionary (**PVSyn**: *auftragen*:*auftischen*), or (iii) because the BVs of two PVs with identical particles are synonymous according to the *Duden* (**BVSyn**: auf*reissen*:auf*platzen*).

Figure 5.6 shows the f-scores for predicting literality and non-literality across the three settings, in comparison to the main experiments (”All”). Regarding ”ALL”: We removed verbs with less than ten sentences, resulting in a list of 151 PVs and

[7] http://www.duden.de

22 650 pairs. The number of PV pairs in the settings and the majority accuracy for these PV pairs are also provided, because the experiment sets differ in size. We can see that PVs with the same BV (**SameBV**) predict each other's classifications well regarding literal but not regarding non-literal sentences. This behavior illustrates the contribution of the particle to the PV meaning: The same BVs with different particles potentially differ strongly, if the particles do not agree on one or more senses. Synonymous PVs (**PVSyn**) predict each other as well in literal as in non-literal cases. Since the PVs in all the cases are supposed to have the same meaning, this behavior is also reasonable. An increase in both literal and non-literal F1 is reached for PV pairs with the same particle and synonymous BVs (**BVSyn**), because the BVs are supposed to carry the same meaning, and the identical particles trigger similar meaning shifts. Overall, the experiment demonstrates that synonymous verbs undergo similar meaning shifts, and that a particle initiates similar meaning shifts when applied to synonymous BVs.

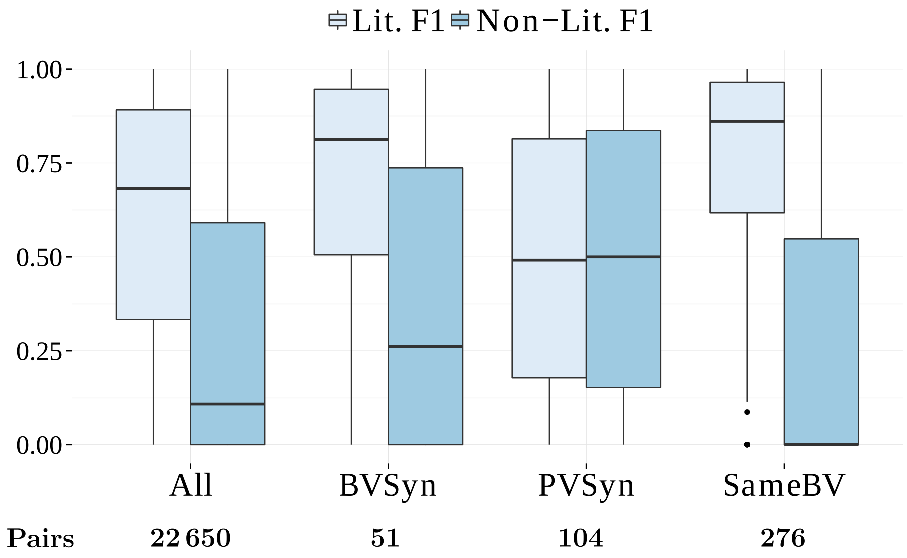


Figure 5.6: Prediction for semantically related PVs.

5.1.5 Feature and Error Analysis

Indicators of Non-Literality: In this part of the work, we perform a qualitative analysis of the most salient features.

Information Gain: First of all, we looked into the feature space by computing the information gain within the best random forest classifier. The information gain (*I-Gain*) provides the improvement in information entropy regarding our feature space and the class labels, as defined by Equation (5.1).

$$\textit{I-Gain}(\textit{Class,Feat}) = H(\textit{Class}) - H(\textit{Class}|\textit{Feat}) \tag{5.1}$$

The information gain does not take feature interaction into account, but determines the importance of the individual features. Applying this method reveals the three most salient features: unigrams (0.31), abstractness ratings of the context nouns (0.17), and distributional fit of the BVs (0.11). Therefore, the information gain confirms our results from the main experiments, where these three features worked best.

In addition, we noticed that for all the features, higher weights were given to dimensions that depend on nouns (such as the common nouns in the PV contexts, and the subject and object nouns), in comparison to proper names, verbs, adjectives, and adverbs. For example, the abstractness ratings of the adverbs were ranked second lowest with a score of 0.005, and the distributional fit between BVs and adjectives was ranked last with a zero score, indicating that this feature provides no additional information for our dataset.

Distributional Fit: We now take a look at the distributional fit feature, the best performing feature in the main experiments, when combined with particle knowledge. Figure 5.7 focusing on the distributional fit between BVs and common nouns (as determined third best by the information gain) confirms that the feature is helpful in distinguishing literal vs. non-literal PV sentences across particles: The medians in the boxplots for literal sentences are clearly above those for non-literal sentences. The plots confirm that BVs can be exploited to identify compositional uses of PVs (which in turn refer to literal usage).

Looking into individual PVs confirms that this feature distinguishes well between the literal and non-literal sentences. On the other hand, we also find PVs where this feature is not able to identify non-literal language use. Figure 5.8 presents the boxplots with cosine values for “aufblühen” *(blossom out)* and “auflodern” (*burn up*), where the feature works well, in comparison to “absaufen” *(to drown)*, where the feature cannot distinguish (non-)literal language. Thus, the feature will only help in cases where a transparent particle verb can be used metaphorically. On the other

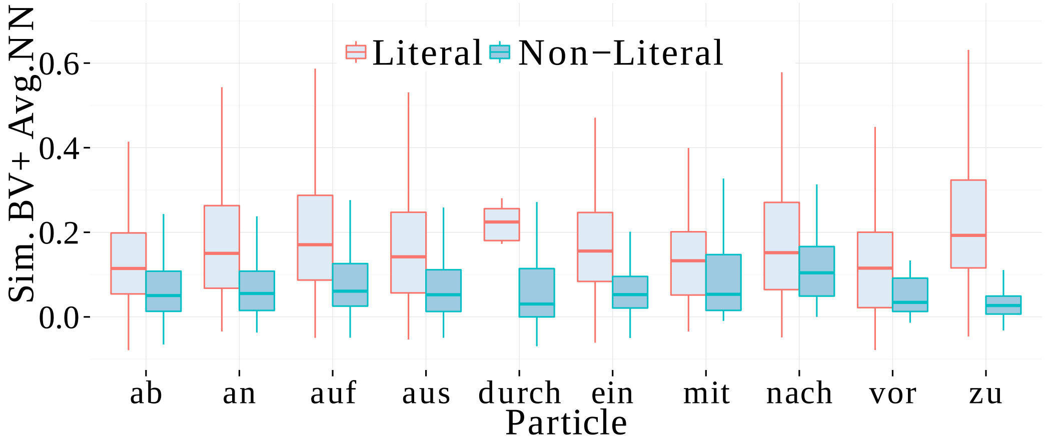


Figure 5.7: Distributional fit of BVs and context nouns in (non-)literal sentences across particles.

hand, it provides little information for lexicalized or non-transparent PVs, since their BVs will always provide a low similarity with context.

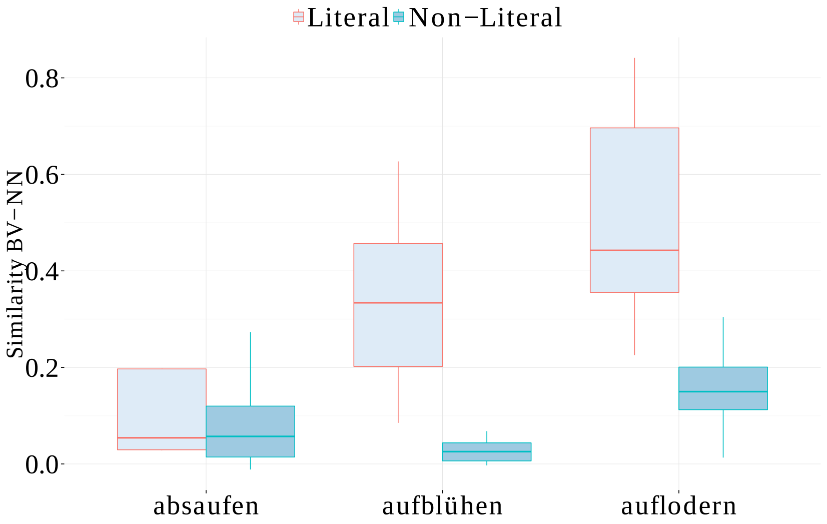


Figure 5.8: Example PVs and their distributional fit of BVs and context nouns in literal and non-literal use.

Abstractness of Context: Finally, we take a look at the abstractness feature, which was also among the best performing features in the main experiments, and which is generally assumed to represent a salient indicator of non-literal language usage. Figure 5.9 focuses on the abstractness of common nouns in the PV sentences[8] (as

[8] High values indicate concreteness.

determined second best by the information gain) and confirms that the feature is also helpful in distinguishing literal vs. non-literal PV sentences across particles: Again, the medians in the boxplots for literal sentences are clearly above those for non-literal sentences. The plots confirm that contextual abstractness is a salient indicator of non-literal language usage.

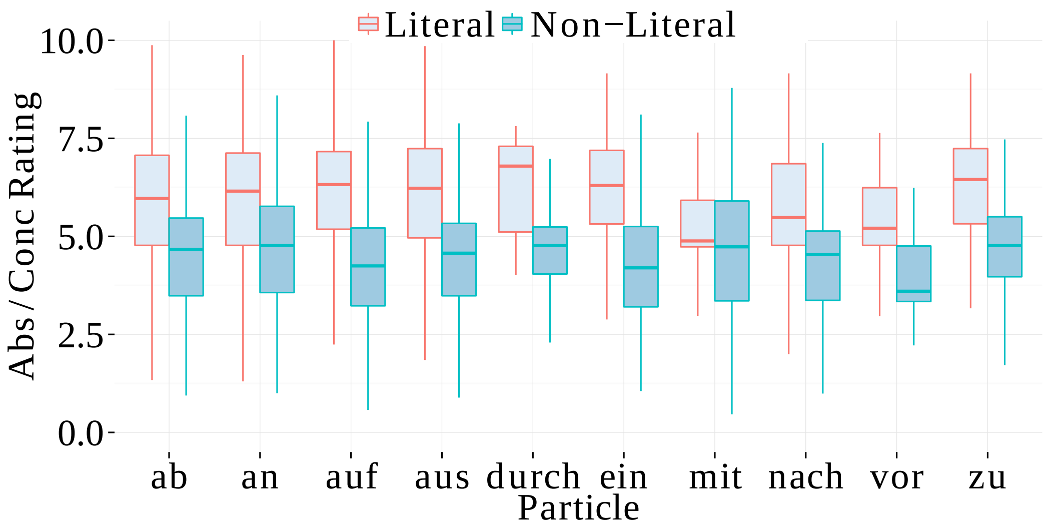


Figure 5.9: Average abstractness ratings of context nouns in (non-)literal sentences across particles.

Looking into individual PVs again confirms that this feature distinguishes well between the literal and non-literal sentences but also that there are PVs where this feature is not able to identify non-literal language use. Figure 5.10 presents the boxplots with abstractness ratings for “anstauen” *(accumulate)* and “durchsickern” *(leak through)*, where the feature works well, in comparison to “antanzen” *(waltz in)* and especially “ausklingen” *(fade/finish)*, where the feature cannot distinguish (non-)literal language usage.

We now focus on false decision. While the AC ratings are beneficial for many PVs, we often observed that these ratings trick the classifier into guessing the wrong class. The classifier assumes a literal sentences when the token based context provides high concreteness (especially noun concreteness) as in the example of ‘antanzen’ *(literal: dancing along, non-lit: to show up)*. Two example sentences where the abstractness feature goes wrong for a good reason are as follows.

1. ”Aber wir sollten doch um fünf zum Essen **antanzen**.”
   *But we should **show up (lit: waltz in)** for dinner at five*

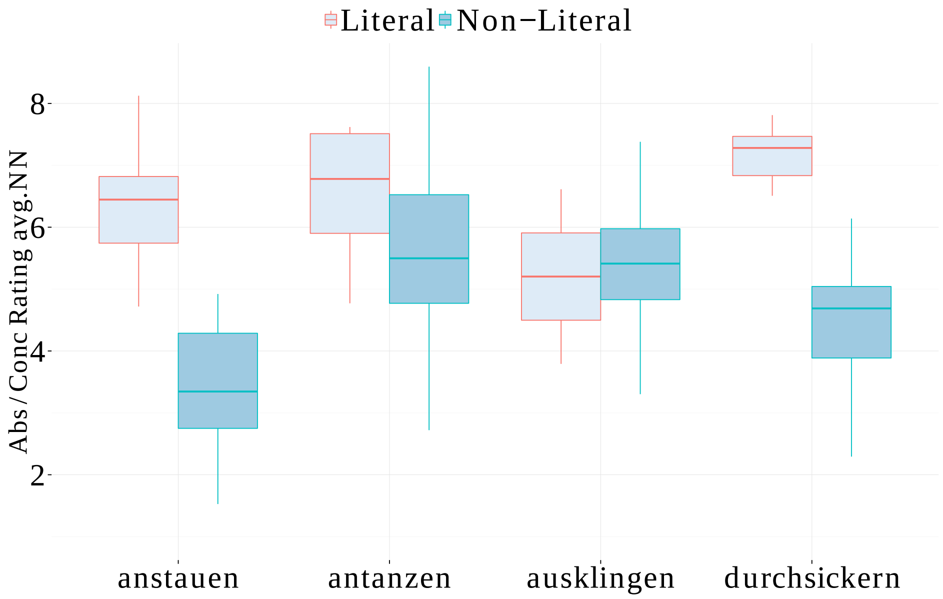


Figure 5.10: Example PVs and their average abstractness ratings of context nouns in (non-)literal use.

2. ”Ich liebe Emotionen, deshalb **summen** alle **mit**.”
   *I love emotions, therefore everyone **hums along***

In (1), the context nouns are concrete (*we; dinner*), but the language usage is non-literal. In contrast, in (2), the object noun in the sentence is highly abstract (*emotion*), but the language usage is literal. These examples illustrate that contextual abstractness is not always a perfect indicator of non-literal language usage.

5.1.6 Summary

Based on our novel literal/non-literal annotated collection of PVs in context, we presented a classifier that predicts literal vs. non-literal language usage for German PVs. This work is the first to address such a classification for this kind of semantically challenging type of multiword expressions. The classifier significantly outperformed the baseline by improving standard features with noun clusters and a PV-specific distributional fit feature. Moreover, we showed that BVs can be exploited to identify compositional uses of PVs. In addition, we illustrated the potential and the limits of the most salient classification features in predicting PV non-literal language usage. PV-specific experiments indicated that PVs whose particles share aspects of ambiguity and which incorporate semantically related BVs seem to undergo similar meaning shifts. The contributions of this work can be summarized as follows:

1. We presented a random forest classifier that correctly identifies 86.8% of literal vs. non-literal language usage within a novel dataset of 6 436 annotated sentences, in comparison to a majority baseline of 64.9%.

2. We successfully incorporated salient PV-specific features and noun clusters in addition to standard bag-of-words features and affective ratings.

3. We demonstrated that PVs with semantically similar particles and semantically similar BVs can predict each others' literal vs. non-literal language usage.

4. We illustrated the potential and the limits of the most salient classification features in predicting PV non-literal language usage.

5. We created a novel public available dataset, annotated for literal vs. non-literal particle verb usage. To the best of our knowledge, this resource is the largest available resource for non-literal German language. We hope that this resource can be used to facilitate further research on German PVs and non-literal language.

5.2 Type-based Multi-Sense Discrimination

5.2.1 Introduction

To date, most distributional semantic models that addressed specific semantic tasks have worked on the type level (e.g., Baroni et al. (2014b), Köper et al. (2015), Levy et al. (2015), Pennington et al. (2014)). In other words, each word lemma is represented by a weighted feature vector, where features typically correspond to words that co-occur in particular contexts. When using word embeddings to overcome the problematic sparsity of word vectors, the models rely on neural methods to represent words as low-dimensional vectors.

In contrast, distributional semantic models that break down word type vectors to word sense vectors, have predominantly be applied to Word Sense Disambiguation/Discrimination or (Cross-lingual) Lexical Substitution (McCarthy and Navigli, 2007; Mihalcea et al., 2010; Jurgens and Klapaftis, 2013).

To our knowledge, there is little work on DSMs that distinguishes between word senses and addresses various semantic relatedness tasks. Among the few exceptions are Li and Jurafsky (2015) who evaluated multi-sense embeddings on semantic relation identification (for nouns only) and semantic relatedness between sentences, and Iacobacci et al. (2015) who applied multi-sense embeddings to word and relational similarity.

In this work, we compare and extend approaches to obtain multi-sense embeddings, to model word senses on the token level. We focus on German verbs, and we evaluate the model variants on various semantic tasks: semantic verb classification; the prediction of compositionality; and the detection of non-literal language usage. While there is no overall best model, all the models significantly outperform a *word2vec* single-sense skip baseline, thus, demonstrating the need to distinguish between word senses in a distributional semantic model.

5.2.2 Multi-Sense Embeddings

We implemented and applied several variants of state-of-the-art methods for obtaining multi-sense embeddings. We avoid using sense learning techniques that rely on a predefined sense inventory, as we assume that most PV senses, and in particular senses from low frequency PVs, are not covered in such resources. Hence, we restrict the selection to models that perform unsupervised and non-parametric sense learning, i.e., methods that learn potentially different numbers of senses per word, using only a corpus but no sense inventory.

(1) Joint learning of sense representations and application of sense disambiguation From this advanced family of multi-sense embedding induction, we applied the non-parametric multiple-sense skip-grams (**NP-MSSG**), cf. Neelakantan et al. (2014), and skip-grams extended by the Chinese Restaurant Process (**ChinRestP**), cf. Li and Jurafsky (2015).

(2) Successive learning of single-sense representations and sense disambiguation This class of approaches also relies on skip-grams but learns senses only in a later stage. Pelevina et al. (2016) introduced a non-parametric method that computes a graph relying on cosine-based nearest neighbors, after learning single-sense representations. The graph-clustering algorithm *Chinese Whispers* (Biemann, 2006) identifies senses in the graph, to induce multi-sense embeddings by applying a composition function to word senses. We refer to this approach as **ChinWhisp**.

(3) Single-sense representations for multi-sense corpus annotations In this class of techniques, multi-sense embeddings are also learned in a two-stage procedure: In a first stage, a corpus is automatically sense-annotated by appending a sense index to every word token (e.g., $apple_1$, $apple_2$, etc.). In a second stage, standard techniques are applied to learn single-sense representations for the annotated senses in the corpus. Since the annotations distinguish between senses, the "single-sense" representations effectively represent multi-sense embeddings. For example, Iacobacci et al. (2015) perform the first step by using an off-the-shelf word sense disambiguation tool, and the second step by applying Mikolov's *word2vec* tool (Mikolov et al., 2013d,c).

We investigate several variants regarding the automatic corpus sense annotation.

1. Rather than applying an off-the-shelf WSD tool, we apply the topic-based sense learning method from (Lau et al., 2012), the Hierarchical Dirichlet process (**HDP**) (Teh et al., 2004). The HDP mixture model is a natural non-parametric generalization of the Latent Dirichlet allocation (Blei et al., 2003), where the number of topics can be unbounded and learned directly from the data. We apply HDP by extracting every sentence for each verb type from our corpus. Then, we train the HDP individually for each verb. In the last training iteration we mark each occurrence of a verb type in the corpus with the number of the topic that provided the largest membership value for the respective sentence and that topic.

2. As an alternative to the topic model, we apply different clustering algorithms, which not only allows more flexibility in the sense classification technique but also regarding the verb features: we represent each verb token by a vector: We look up the individual vector representations of the verb's context words, and create the verb token vector as the average vector of these context words, ignoring the target verb. This simple kind of phrase/sentence representation has been shown to work well on a variety of tasks (e.g., Milajevs et al. (2014), Hill et al. (2016)). In addition, it allows us to compare different types of context features: (a) all the nouns in the sentence (NN), and (b) all the words in a symmetrical window of size 10, weighted by the exponential decay function (w10EXP), cf. Iacobacci et al. (2016). In version (b), immediate surrounding

words contribute 10 times more to the average vector than the most distant words in the window.

For the actual clustering, we compare non-parametric flat and hierarchical methods. Regarding the HDP, we cluster verb tokens separately and, then, we mark each verb token with a tag corresponding to a cluster number. The number of clusters containing a specific verb type corresponds to its number of senses. For flat clustering, we use **X-Means** (Pelleg and Moore, 1999), which extends the standard hard k-means clustering approach into a non-parametric soft clustering. For hierarchical clustering, we use *balanced iterative reducing and clustering using hierarchies* **BIRCH** (Zhang et al., 1996), see Section 3.2.1 for background information on these techniques. Note that for all of these approaches, we can also keep track of the number of times a certain sense was assigned; we use this information to compute a weight value for each sense.

It shall be mentioned that the idea to use clustering to obtain sense representations is not new. In fact older approaches such as Clustering by Committee (Pantel, 2003) or the multi-prototype approaches (Reisinger and Mooney, 2010; Huang et al., 2012) also build meaning vectors from the disambiguated words.

5.2.3 Experiments

Corpus & Target Verbs As a corpus resource for our target verbs as well as for the experimental setup, we use *DECOW14AX* (Section 3.5). Based on the morphological annotation, we extracted the lemmas of all the verb types from the corpus with frequencies >100 (regarding BVs) and >200 (regarding complex verbs), and all their sentence contexts.

The total selection of German verb types contains 11 869 lemmas, including 6 998 complex verbs. The PVs range across 13 particles: ab, an, auf, aus, durch, ein, mit, nach, über, um, unter, vor, and zu. Of these 11 869 target verbs, the German standard dictionary *DUDEN*[9] covers only 7 224 (61%), showing that our verb data includes a much larger verb inventory.

Experiment Setup The different models have multiple parameters. The preliminary experiments showed that some multi-sense approaches cannot deal with large vocabularies. Therefore, we set the initial vocabulary to the 200K most frequent

[9] www.duden.de

word types, without removing any of the target verb types. The maximum number of senses per verb type was set to 20. We enabled the multi-sense learning only for our target verbs while all other words obtain only a single sense per model. Regarding the skip-gram architecture, we relied on a symmetrical window of size 10, negative sampling with 15 samples, vector dimensionality of 400 and one corpus iteration. Regarding the HDP, we removed words that appeared only once, and words that appeared in more than 10% of the sentences. Furthermore, we used a maximum of 5 000 randomly chosen contexts to learn the initial topics (HDP), centroids (X-Means), or trees (BIRCH), due to the high-dimensional representations of the sentences. All other individual model-specific parameters were set to the default. Our baseline model is a single-sense skip-gram model as obtained by *word2vec*.

Implementations For the HDP, we relied on the python implementation from *gensim*[10]. For X-Means, we used the java implementation *ClodHopper*[11]. For BIRCH we used the java implementation *JBIRCH*[12].

5.2.4 Evaluation

For assessing the quality of multi-sense embeddings, most of the work have focused on predicting similarity and relatedness of word pairs, comparing the predictions against human judgments. This is either done for single words out of context, e.g., by using WordSim-353 (Finkelstein et al., 2002), or by predicting a similarity score for words in context relying on Stanford's Contextual Word Similarities dataset Huang et al. (2012). A popular evaluation choice is also the usage of word sense induction and disambiguation shared tasks, such as SemEval-2007 Task 2 (Agirre and Soroa, 2007), SemEval-2010 Task 14 (Manandhar and Klapaftis, 2009), and the SemEval-2013 Task 13 (Jurgens and Klapaftis, 2013). However, most of these resources are available for the English language only.

In this work we focus on German, therefore we evaluate our models on various available semantic tasks: general predictions of semantic similarity, and specific tasks regarding complex German verbs, i.e. semantic classification; prediction of compositionality; detection of non-literal language usage. The goal of the evaluation is to

[10] `https://radimrehurek.com/gensim/models/hdpmodel.html`

[11] `https://github.com/rscarberry-wa/clodhopper`

[12] `https://github.com/perdisci/jbirch`

explore whether the distinction of verb senses in our multi-sense embedding models leads to an improvement of model predictions across semantic tasks.

Similarity Traditionally, distributional word representations are predominantly evaluated on their ability to predict the degree of similarity for word pairs in existing benchmarks. The predicted degrees of similarity are compared against human similarity ratings. For our German targets, we use the German versions of *WordSim-353* and *SimLex-999* (Leviant and Reichart, 2015). Across the following experiments, we predict cosine similarity for multi-sense embeddings by computing a sense-weighted average vector for each word. To assess the predictions, we compare them against the human gold-standard judgment scores using Spearman's Rank-Order Correlation Coefficient ρ (Siegel and Castellan, 1988). We conducted preliminary experiments, using different similarity techniques, such as a none-weighted average or define similarity of two words as the similarity of their closest senses. We found that sense-weighted average vector performs best, which confirms the findings from psychological studies (Tversky, 1977; Markman and Gentner, 1993). According to their findings, while judging semantic similarity of a pair of words, humans consider different meanings words and not only the predominant or closest ones.

Table 5.4 presents the results. For this general semantic task, the multi-sense embeddings do not provide significant improvements. The best results are achieved by ChinRestP for *GerSimLex* and x-Means(w10Exp) for *GerWS353*, but these results are close to the baselines.

Model	GerWS353	GerSimLex
NP-MSSGR	.62	.42
ChinRestP	.64	**.46**
ChinWhisp	.64	.36
HDP	.63	.45
x-Means(NN)	.64	.43
x-Means(w10Exp)	**.65**	.44
BIRCH(NN)	.63	.44
BIRCH(w10Exp)	.64	.45
Baseline	**.65**	.45

Table 5.4: Results for the word similarity datasets. Numbers show Spearman's ρ.

Compositionality Addressing the compositionality of complex words is a crucial ingredient for lexicography and NLP applications, to know whether the expression should be treated as a whole, or through its constituents, and what the expression means.

In this evaluation, we predict the degree of compositionality of German PVs, i.e., the degree of relatedness between a complex verb and its corresponding base verb (such as *abnehmen–nehmen* "take over–take", and *anfangen–fangen* "begin–catch"). The predictions are evaluated against the GhostPV dataset (Bott et al., 2016) and PV150 (Bott and Schulte im Walde, 2015). Note that our setup, and in particular our baseline, is comparable with the systematic comparison of DSMs performed in Section 4.1. The only noteworthy difference, is that the vector space in this setup contains only 200K words. Table 5.5 presents the results. ChinWhisp performs significantly better than the baseline, while most other models are performing equally to or even inferior to the baseline.

Model	GhostPV	PV150
NP-MSSGR	.20	26
ChinRestP	.30	29
ChinWhisp	**.32**	**33**
HDP	.19	21
x-Means(NN)	.19	22
x-Means(w10Exp)	.26	26
BIRCH(NN)	.28	29
BIRCH(w10Exp)	.26	28
Baseline	.26	28

Table 5.5: Results for predicting compositionality for GhostPV and PV150. Numbers show Spearman's ρ.

Semantic Verb Classification Semantic verb classifications are of great interest to NLP, specifically regarding the pervasive problem of data sparseness in the processing of natural language. Such classifications have been used in applications such as *word sense disambiguation* (Dorr and Jones, 1996; Kohomban and Lee, 2005; McCarthy et al., 2007), *parsing* (Carroll et al., 1998; Carroll and Fang, 2004), *machine translation* (Prescher et al., 2000; Koehn and Hoang, 2007; Weller et al., 2014), and *information extraction* (Surdeanu et al., 2003; Venturi et al., 2009).

We target the semantic classification of German complex verbs by applying hard clustering to multi-sense embeddings, rather than using soft clustering. Focusing on PVs across three particles (*ab, an, auf*), we aim to obtain cluster analyses that resemble existing manual sense classifications based on formal semantic definitions (Kliche, 2011; Lechler and Roßdeutscher, 2009b; Springorum, 2011). All the datasets represent fuzzy gold standards where some verbs belong to more than one class. Since the original resources are very imbalanced and fine grained, we asked Sylvia Springorum to reduce and merge classes to their most important ones. The resulting classes are provide in the appendix of this thesis (Section 8). The "ab" classification contains 9 classes; the "an" classification contains 8 classes; the "auf" classification contains 11 classes. "*All*" refers to the concatenation of all tasks.

Using multi-sense embeddings in a hard clustering (rather than single-sense embeddings in a soft clustering) avoids the usage of a cluster membership threshold, which most soft clustering algorithms require. In contrast, the clustering algorithm outputs a membership degree for each element and each cluster, i.e., a fuzzy membership. We rely on k-Means for clustering our multi-sense embeddings, and compare against a fuzzy c-Means baseline with single-sense embeddings. (using every possible threshold within a range of $[0.01, 0.99]$ to determine the memberships, and reporting the one providing the highest score). As an evaluation measure, we relied on *B-Cubed* (Bagga and Baldwin, 1998); see Section 3.2.2, and report f-score between the soft extension of precision and recall.

Table 5.6 presents the results. Overall, ChinRestP works best, and ChinWhisp and the BIRCH variants work similarly well. NP-MSSGR is worst. A manual inspection revealed that NP-MSSGR tends to learn too many representations per verb and, therefore, assigns many verbs to multiple clusters, resulting in too large and fuzzy clusters.

Model	*ab*	*an*	*auf*	all
NP-MSSGR	.12	.18	.15	.05
ChinRestP	.24	**.31**	.27	**.13**
ChinWhisp	**.26**	.30	**.28**	.11
HDP	.24	.28	.25	.10
x-Means(NN)	.17	.25	.18	.09
x-Means(w10Exp)	.17	.24	.20	.09
BIRCH(NN)	**.26**	.30	.26	.12
BIRCH(w10Exp)	**.26**	.32	.25	.12
Baseline	.25	.26	.19	.11

Table 5.6: Results for semantic classification. Numbers show (fuzzy) B-Cubed F_1

Detecting Non-Literal Meaning We explore the prediction of literal vs. non-literal language usage of German complex verbs, relying on an existing dataset containing 159 PVs within 6 436 sentences (Köper and Schulte im Walde, 2016). Thus we rely on the dataset from Section 2.3. Each sentence is annotated on literal vs. non-literal language usage, comprising 4 174 literal and 2 262 non-literal uses across the 159 PVs.

We applied the same experimental setup using ten-fold cross validation and the Multinomial Naive Bayes (MNB). Furthermore, we re-implemented their system as a baseline, using bag-of-words unigram context features, and added sense information based on the embeddings. For a given sentence, we compare which sense vector fits best to the specific context. This is done by computing a cosine-similarity score between a verb sense vector $verb_i$ and the vectors of all context words in the sentence. Then, we add a verb-sense specific token based on the most similar sense embedding to the unigram list. The underlying assumption is that a specific sense is used either in literal or in non-literal usage. When feeding the training data to the classifier, it should, thus, automatically assign a high probability for features that predominantly occur for the respective classes. Using this method we automatically penalize methods that assign too many senses to verbs, because the more senses we add, the lower the probability is that we observe the same sense again on a test instance. The other extreme case would add the same sense to every instance; here, we would basically learn nothing since the sense would appear for both classes: literal and non-literal.

A major difference between our setup and the one by Köper and Schulte im Walde (2016) (Section 2.3) is the information about the verb itself. In our experiments, the classifier has knowledge about the verb in a sentence, while in the previous setup the verb has been removed, to avoid learning a verb-specific majority baseline (since some verbs have only literal/non-literal sentences). For this reason, our baseline (i.e., one sense per verb) is already higher than their reported baseline. The remaining parts of our experimental setting are, however, done as by Köper and Schulte im Walde (2016). To evaluate the classifiers, we calculate the f-score for both classes and accuracy.

Table 5.7 shows the results. All the multi-sense embedding models clearly outperform the single-sense baseline model. The overall best models are the clustering models x-Means and BIRCH.

Looking at the sentences, we found remarkable performance for some verbs. Our best method assigned sense_4 of "ausmalen" *(literal: to paint sth., metaphorical :to describe)* exclusively to 15 metaphorical usages co-occurring with words such as "Ereignis" *(event)*, "Folge" *(consequence)* or "Geld" *(money)*. Literal sentences, co-occurring next to "Bild" *(picture)*, "Pinsel" *(brush)* or "Wand" *(wall)* where mostly labeled with

sense_1, allowing the classifier to learn that these features are valuable for the classification task.

Model	Lit. F_1	Non-Lit. F_1	Acc
NP-MSSGR	92.4	84.9	89.8
ChinRestP	92.0	84.1	89.3
ChinWhisp	92.7	85.4	90.3
HDP	92.6	85.1	90.1
x-Means(NN)	**94.1**	**88.2**	**92.1**
x-Means(w10Exp)	93.2	86.3	90.9
BIRCH(NN)	93.1	86.2	90.8
BIRCH(w10Exp)	93.3	86.7	91.1
Baseline (K&SiW)	89.8	76.5	85.7

Table 5.7: Results for non-literal language.

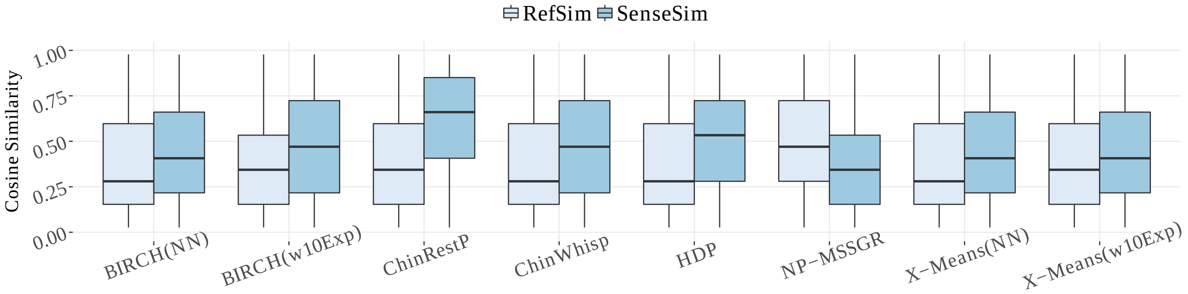


Figure 5.11: Cosine similarity between all sense pairs within a specific embedding model: many senses are highly similar to each other. RefSim refers to similarity distribution over reference set of highly similar word pairs.

5.2.5 Discussion & Summary

Overall, our experiments demonstrated that the variants of multi-sense embeddings we applied across semantic tasks are successful in comparison to single-sense baselines. In all the tasks we presented, some, most, or even all of the multi-sense embeddings outperformed the single-sense baselines, thus, demonstrating the need to distinguish between word senses in a distributional semantic model.

The best multi-sense embeddings varied across the semantic tasks. In other words, there was no type of multi-sense embedding that performed superior to all other multi-sense embedding types. Even ChinWhisp, which was among the most successful embeddings across many tasks, exhibited a weakness on one task (i.e., similarity for German SimLex). Furthermore, we showed that multi-sense embeddings can be potentially useful for the detection of non-literal language and for semantic soft-clustering. On the contrary, the usage of multi-sense embeddings for estimating similarity or compositionality via cosine-similarity requires a single vector. Combining vectors into a single (mean/sense-weighted mean) is essentially not different from default monosemous vector spaces.

We also looked into the inter-sense similarity within the embedding models. Figure 5.11 presents box-plots on the cosine similarity between all sense pairs within a specific embedding model. To put this similarity distribution into context, we provide a reference distribution (RefSim) per vector space. The reference is built by taking the 20% most similar word pairs from the German SimLex (Leviant and Reichart, 2015). Thus, we expect high similarity for this reference set. The plot shows that overall, the identified senses in the models are quite similar to each other. The strongest inter-sense similarity can be found for ChinRestP. Only the topic modeling approach (HDP) exhibits a distribution where the senses are less similar to each other.

Looking into the embeddings across multi-sense approaches, we found that –even though the embeddings were trained on the same data– the average number of senses differs strongly across the embedding models: NP-MSSGR, ChinRestP, and ChinWhisp have an average number of less than 2 senses per word, while the x-Means and BIRCH models have an average number between 3.2 and 7.6 senses. Most of the senses are obtained by the HDP (15.4); however, many senses received little weight.

This diversity of success across embedding types and semantic tasks demonstrates that an evaluation of semantic models on a general task such as semantic similarity is not sufficient.

5.3 Chapter Summary

This chapter presented experiments on particle-verb senses. We predicted literal and non-literal usage in context and presented models to learn sense-specific representations.

We created and released a large resource, annotated for literal vs. non-literal particle-verb usage. Using this resource, we could successfully combine a wide set of different features to obtain high accuracy for non-literal language detection. Among others, we relied on our novel large resource of automatically created German abstractness norms as well as PV-specific feature that makes use of contextual compositionality information (distributional-fit). Furthermore, particle and PV-specific experiments confirmed that PVs with semantically similar BVs can predict each others literal vs. non-literal language usage.

Our experiments on sense-specific representation learning includes a comparison of novel and existing techniques applied across PV-specific semantic tasks. Although none of the methods performed well across all tasks, we observed that in many cases the multi-sense representations outperform a single-sense default representation.

6 Regular Meaning Shifts

The previous chapter presented computational approaches to model PV senses and non-literal language of PVs. The modeling was done on a type-level for senses and on a token-level for non-literal language.

Beyond PV senses and their non-literal language usage, this section studies patterns and regularities in meaning shifts. Section 6.1 represents a novel sentence collection that allows us to study changes from the source domain to the target domain.

Section 6.2 presents a type-based and supervised computational model of analogy. Here, we look at pairs of BV-PV combinations to detect various kinds of meaning shifts.

The experiments described in this chapter are published in Köper and Schulte im Walde (2018) and Schulte im Walde et al. (2018).

6.1 Token-Based Regular Meaning Shifts across Domains: Source→Target

Meaning shifts are typically represented as mapping from a rather concrete source-domain meaning to a rather abstract target-domain meaning (Lakoff and Johnson, 1980). For example, the abstract conceptual domain Time may be illustrated in terms of the structurally similar, more concrete domain Money (see Section 2.3.1), enabling non-literal language such as *to save time* and *to spend time*.

For German PVs, meaning shifts frequently take place when combining a BV from a concrete source domain with a particle, as, for example, "abschminken". The PV abschminken has a literal meaning (*to remove make-up*) and a shifted, non-literal meaning (*to forget about something*). The BV "schminken" is taken from the domain Human Body, resulting in a PV meaning (possibly among other meanings) related to an abstract target domain such as Desire.

In this section, we exploit common source–target domain combinations for German PVs. Hence, we present a new collection for German PVs. The dataset includes 138 German BVs and their 323 existing PVs with particle prefixes *ab, an, auf, aus*. For all the target verbs, we collected sentences from 15 human participants across a specified set of domains, to address their ambiguity in context. In this way, our dataset offers source–target domain combinations for assessing BV–PV meaning shifts across PVs and particle types.

It should be mentioned, that the resource also contains spatial directionality, e.g., each sentence was labeled with a direction arrow (↑, ↓, ←, →). As the focus of this thesis is the mapping from source to target domains, we ignore the directionality information in this description.

6.1.1 Data Collection Target Verbs, Domains

In this section, we describe our selections and representations of BV and PV targets, as well as the source and target domains.

German BVs and PVs: Based on the source-domain descriptions by Kövecses (2002), below, we identified BVs that (i) supposedly belong to the respective source domain, and (ii) we expected to undergo meaning shifts when combined with one of our target particle types.

All of the BVs were systematically combined with the four prefix particles *ab, an, auf, aus*, resulting in a total of 552 PVs. Since we did not want to include neologisms into our PV targets, we then checked the PV existence in the online version of the German dictionary, *Duden*[1]. The final list of target PVs that were found in the dictionary comprised 323 verbs.

Domains of Meaning Shift: Our source and target domains were taken from specifications in Kövecses (2002), which we assumed to ensure a more stratified and generally applicable set of domains involved in meaning shifts. Table 6.1 lists all 13 source and 12 target domains by Kövecses (2002). Note that we used German translations to compile our collection. Regarding the source domains, we added one domain to Kövecses' original list, i.e., SOUND which we expected to play a role in BV–PV meaning shifts Springorum et al. (2013b).

[1] www.duden.de/suchen/dudenonline/

Annotation Instructions: We randomly distributed BVs and PVs over lists with 35 verbs each. The annotators were asked

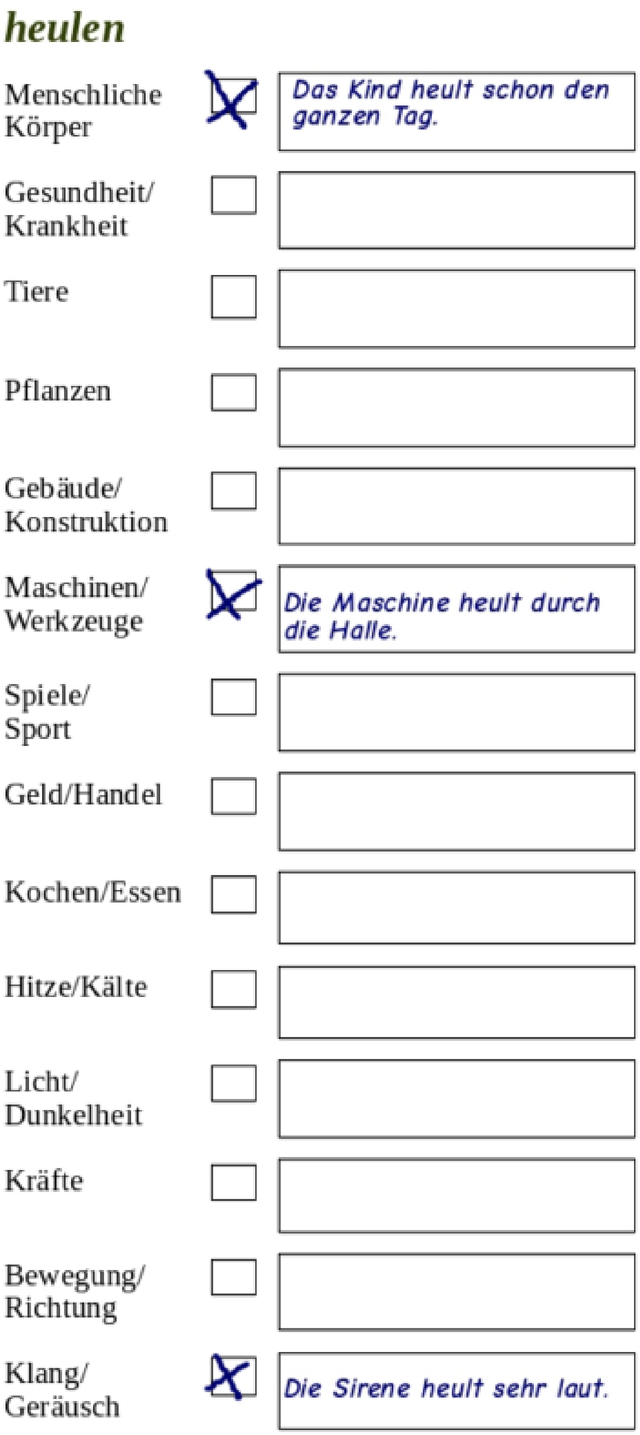
1. To choose one or more pre-defined semantic domain classes for each verb,
2. To provide an example sentence to illustrate the class assignment

The classes (i.e., the source domains in the BV lists, and the target domains in the PV lists) were described by key words (e.g., the German equivalents of *appearance, growth, cultivation, care, use* for the source domain Pflanzen "Plants"). Then, the annotators were provided with one example annotation (cf. Figure 6.1 for the verb "heulen" *to howl*) before they started the annotation process.

Qualitative Description: Table 6.1 shows the total number of sentences that were generated by the participants, and the proportions per domain. In total, we collected 2933 sentences across the 138 BVs and the 14 source domains, and 4487 sentences across the 323 PVs and the 12 target domains. The collection comprises ≥10 sentences per verb for 134 out of 138 BVs (97%), and for 277 out of 323 PVs (86%) The distribution of source domain sentences across domains ranges from a proportion of 3.41% for the domain Forces up to 14.69% for the domain Human Body. The distribution of target domain sentences is more skewed, ranging from 0.47% for the domain Religion up to 33.88% for the domain Event/Action.

Source Domains	No. of Sentences		Target Domains	No. of Sentences	
Human Body	431	14.69%	Event/Action	1,520	33.88%
Animals	322	10.98%	Economy	460	10.25%
Health/Illness	251	8.56%	Emotion/Feeling	452	10.07%
Machines/Tools	242	8.25%	Human Relationships	383	8.54%
Games/Sports	211	7.19%	Life/Death	365	8.13%
Cooking/Food	210	7.16%	Time	292	6.51%
Plants	207	7.06%	Thought	284	6.33%
Economic Transaction	190	6.48%	Communication	280	6.24%
Buildings/Construction	167	5.69%	Society/Nation	181	4.03%
Sound	165	5.63%	Desire	150	3.34%
Heat/Cold	156	5.32%	Morality	99	2.21%
Movement/Direction	154	5.25%	Religion	21	0.47%
Light/Darkness	127	4.33%			
Forces	100	3.41%			
Total:	2,933	100.00%	*Total:*	4,487	100.00%

Table 6.1: Source and target domains: Number and proportions of generated sentences per domain.



heulen

Menschliche Körper	☒	Das Kind heult schon den ganzen Tag.
Gesundheit/ Krankheit	□	
Tiere	□	
Pflanzen	□	
Gebäude/ Konstruktion	□	
Maschinen/ Werkzeuge	☒	Die Maschine heult durch die Halle.
Spiele/ Sport	□	
Geld/Handel	□	
Kochen/Essen	□	
Hitze/Kälte	□	
Licht/ Dunkelheit	□	
Kräfte	□	
Bewegung/ Richtung	□	
Klang/ Geräusch	☒	Die Sirene heult sehr laut.

Figure 6.1: Example annotation for the verb "heulen" *to howl* with (i) a selection of three source domain classes, and (ii) the corresponding three sentences.

6.1.2 Analyses of Meaning Shifts

We now take the first steps into analyzing non-literal language and meaning shifts within our collection. We started out by assuming that *"meaning shifts for German PVs frequently take place when combining a BV from a concrete source domain with a particle, resulting in a PV meaning (possibly among other meanings) related to an abstract target domain"*. Consequently, the generated PV sentences are expected to (i) represent shifted, non-literal language meanings and (ii) exhibit abstract meanings, both considerably more often than the generated BV sentences.

(Non-)Literal BV/PV Language Usage We asked three German native speakers to annotate the 2 933/4 487 BV/PV sentences with ratings on a 6-point scale [0,5], ranging from clearly literal (0) to clearly non-literal (5) language. We obtained a fair agreement on the full scale with a Fleiss' $\kappa = 0.27$. Dividing the scale into two disjunctive ranges [0, 2] and [3, 5] broke down the ratings into binary decisions. Table 6.2 shows the numbers and proportions of BV/PV sentences that were annotated as literal vs. non-literal language usage, distinguishing between full agreement (i.e., all the annotators agreed on the binary category) and majority agreement (i.e., at least two out of the three annotators agreed on the binary category). Agreement on the binary classes reached a moderate[2] Fleiss' $\kappa = 0.47$. We can see that the proportions of non-literal sentences are indeed considerably larger for PVs than for BVs (14.8% vs. 3.2% for full agreement, and 29.5% vs. 14.8% for majority agreement), thus indicating a stronger non-literal language potential for German PVs in comparison to their BVs. Contrary to our assumptions, the participants in the generation

		literal		non-literal	
BVs	full	2,443	83.3%	94	3.2%
	maj	2,674	91.2%	259	8.8%
PVs	full	2,174	48.5%	666	14.8%
	maj	3,150	70.2%	1,337	29.5%

Table 6.2: (Non-)literal language usage in generated BV/PV sentences.

[2] The task and annotation procedure was identical to the one reported in Section 5.1.2. But this collection obtained a lower Fleiss' κ score. To review, the procedure detailed in Section 5.1.2 reports a substantial agreement with a Fleiss'$\kappa = 0.70$ for the binary case. We assume the difference is due to the fact that the annotators for the dataset in Section 5.1.2 were post-docs/PhD students with experience and prior work on MWEs. Hence, it is likely that these annotators had a more coherent understanding of literal and non-literal language. In comparison, this annotation was carried out by Master-students with (presumably) less prior knowledge.

experiment also produced a large number of literal sentences for PVs. In our opinion this indicates (a) the ambiguity of German PVs, which led the participants to refer to literal as well as non-literal senses; and (b) that the presumably strongly abstract target domain definitions did not necessarily enforce non-literal senses.

Abstractness in BV/PV Sentences As meaning shifts typically take place as a mapping from a source domain to a target domain, where the target domain is supposedly more abstract than the source domain, we expect our sentences in the target domains to be more abstract than those in the source domains. Figure 6.2 shows that this is the case:

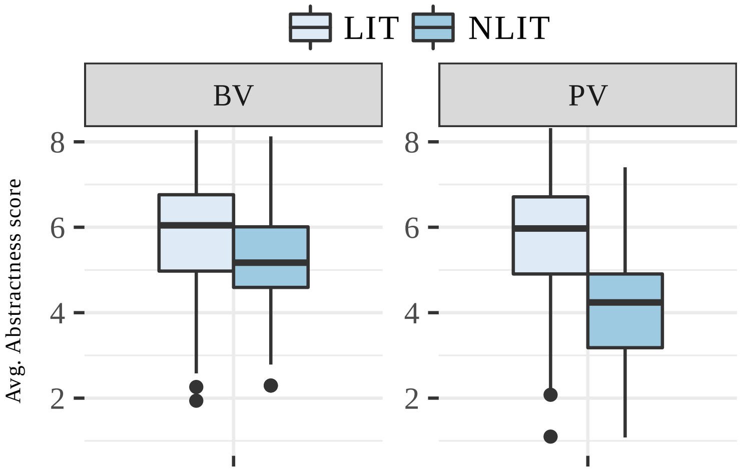
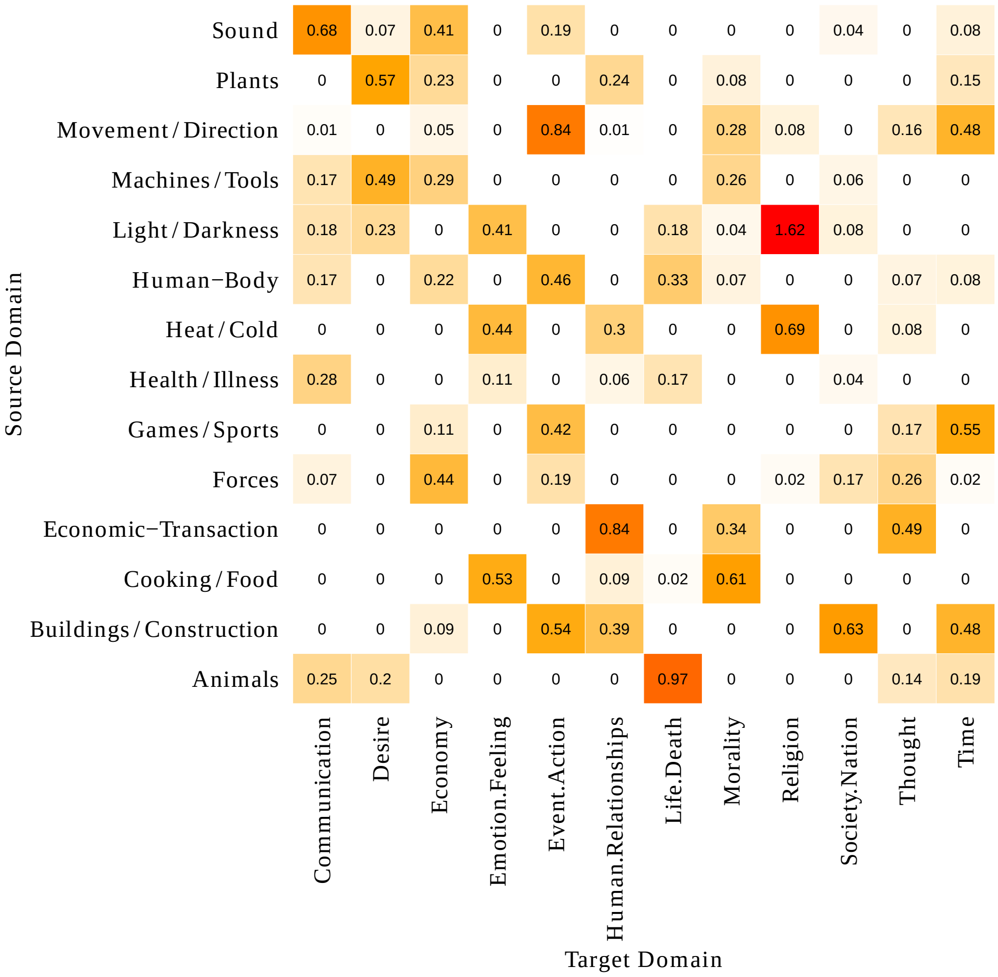


Figure 6.2: Average concreteness of nouns in BV/PV sentences. Low values indicate abstractness and high values indicate concreteness.

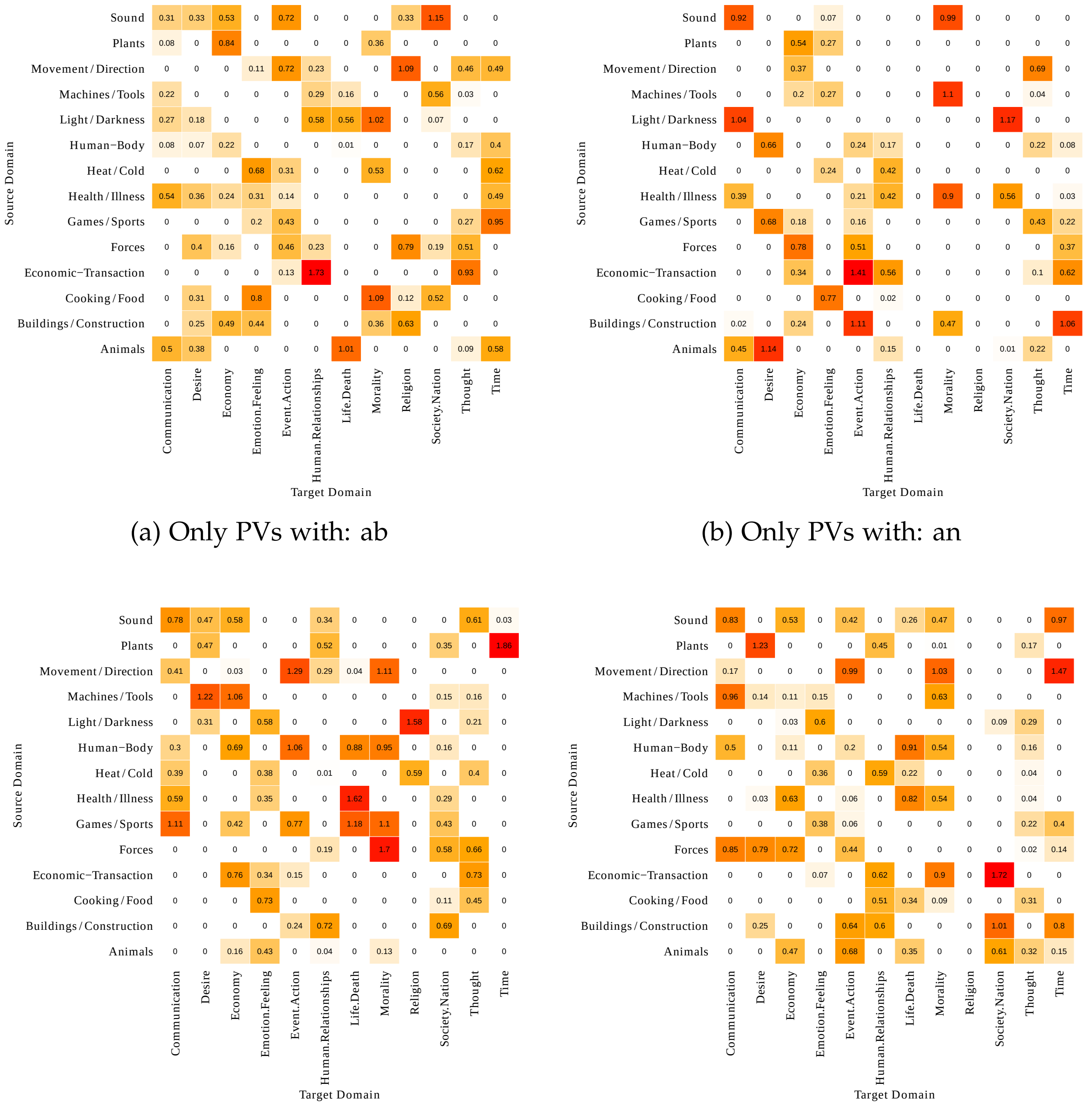
Relying on abstractness/concreteness ratings of a semi-automatically created database (Köper and Schulte im Walde, 2016), we looked up and averaged over the ratings of all nouns in a sentence. The ratings range from 0 (very abstract) to 10 (very concrete). We can see that across verbs, the literal sentences are more concrete than the non-literal sentences. In addition, we can see that the differences in abstractness are much stronger for the PV target-domain sentences than for the BV source-domain sentences.

Source–Target Domain Meaning Shifts Figure 6.3 presents meaning shifts as the strengths of relationships between source and target domains, when looking at only



	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	0.68	0.07	0.41	0	0.19	0	0	0	0	0.04	0	0.08
Plants	0	0.57	0.23	0	0	0.24	0	0.08	0	0	0	0.15
Movement/Direction	0.01	0	0.05	0	0.84	0.01	0	0.28	0.08	0	0.16	0.48
Machines/Tools	0.17	0.49	0.29	0	0	0	0	0.26	0	0.06	0	0
Light/Darkness	0.18	0.23	0	0.41	0	0	0.18	0.04	1.62	0.08	0	0
Human−Body	0.17	0	0.22	0	0.46	0	0.33	0.07	0	0	0.07	0.08
Heat/Cold	0	0	0	0.44	0	0.3	0	0	0.69	0	0.08	0
Health/Illness	0.28	0	0	0.11	0	0.06	0.17	0	0	0.04	0	0
Games/Sports	0	0	0.11	0	0.42	0	0	0	0	0	0.17	0.55
Forces	0.07	0	0.44	0	0.19	0	0	0	0.02	0.17	0.26	0.02
Economic−Transaction	0	0	0	0	0	0.84	0	0.34	0	0	0.49	0
Cooking/Food	0	0	0	0.53	0	0.09	0.02	0.61	0	0	0	0
Buildings/Construction	0	0	0.09	0	0.54	0.39	0	0	0	0.63	0	0.48
Animals	0.25	0.2	0	0	0	0	0.97	0	0	0	0.14	0.19

Figure 6.3: (Literal) source domain → (non-literal) target domain shifts across all particles.



(a) Only PVs with: ab

	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	0.31	0.33	0.53	0	0.72	0	0	0	0.33	1.15	0	0
Plants	0.08	0	0.84	0	0	0	0	0.36	0	0	0	0
Movement / Direction	0	0	0	0.11	0.72	0.23	0	0	1.09	0	0.46	0.49
Machines / Tools	0.22	0	0	0	0	0.29	0.16	0	0	0.56	0.03	0
Light / Darkness	0.27	0.18	0	0	0	0.58	0.56	1.02	0	0.07	0	0
Human−Body	0.08	0.07	0.22	0	0	0	0.01	0	0	0	0.17	0.4
Heat / Cold	0	0	0	0.68	0.31	0	0	0.53	0	0	0	0.62
Health / Illness	0.54	0.36	0.24	0.31	0.14	0	0	0	0	0	0	0.49
Games / Sports	0	0	0	0.2	0.43	0	0	0	0	0	0.27	0.95
Forces	0	0.4	0.16	0	0.46	0.23	0	0	0.79	0.19	0.51	0
Economic−Transaction	0	0	0	0	0.13	1.73	0	0	0	0	0.93	0
Cooking / Food	0	0.31	0	0.8	0	0	0	1.09	0.12	0.52	0	0
Buildings / Construction	0	0.25	0.49	0.44	0	0	0	0.36	0.63	0	0	0
Animals	0.5	0.38	0	0	0	0	1.01	0	0	0	0.09	0.58

(b) Only PVs with: an

	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	0.92	0	0	0.07	0	0	0	0.99	0	0	0	0
Plants	0	0	0.54	0.27	0	0	0	0	0	0	0	0
Movement / Direction	0	0	0.37	0	0	0	0	0	0	0	0.69	0
Machines / Tools	0	0	0.2	0.27	0	0	0	1.1	0	0	0.04	0
Light / Darkness	1.04	0	0	0	0	0	0	0	0	1.17	0	0
Human−Body	0	0.66	0	0	0.24	0.17	0	0	0	0	0.22	0.08
Heat / Cold	0	0	0	0.24	0	0.42	0	0	0	0	0	0
Health / Illness	0.39	0	0	0	0.21	0.42	0	0.9	0	0.56	0	0.03
Games / Sports	0	0.68	0.18	0	0.16	0	0	0	0	0	0.43	0.22
Forces	0	0	0.78	0	0.51	0	0	0	0	0	0	0.37
Economic−Transaction	0	0	0.34	0	1.41	0.56	0	0	0	0	0.1	0.62
Cooking / Food	0	0	0	0.77	0	0.02	0	0	0	0	0	0
Buildings / Construction	0.02	0	0.24	0	1.11	0	0	0.47	0	0	0	1.06
Animals	0.45	1.14	0	0	0	0.15	0	0	0	0.01	0.22	0

(c) Only PVs with: auf

	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	0.78	0.47	0.58	0	0	0.34	0	0	0	0	0.61	0.03
Plants	0	0.47	0	0	0	0.52	0	0	0	0.35	0	1.86
Movement / Direction	0.41	0	0.03	0	1.29	0.29	0.04	1.11	0	0	0	0
Machines / Tools	0	1.22	1.06	0	0	0	0	0	0	0.15	0.16	0
Light / Darkness	0	0.31	0	0.58	0	0	0	0	1.58	0	0.21	0
Human−Body	0.3	0	0.69	0	1.06	0	0.88	0.95	0	0.16	0	0
Heat / Cold	0.39	0	0	0.38	0	0.01	0	0	0.59	0	0.4	0
Health / Illness	0.59	0	0	0.35	0	0	1.62	0	0	0.29	0	0
Games / Sports	1.11	0	0.42	0	0.77	0	1.18	1.1	0	0.43	0	0
Forces	0	0	0	0	0	0.19	0	1.7	0	0.58	0.66	0
Economic−Transaction	0	0	0.76	0.34	0.15	0	0	0	0	0	0.73	0
Cooking / Food	0	0	0	0.73	0	0	0	0	0	0.11	0.45	0
Buildings / Construction	0	0	0	0	0.24	0.72	0	0	0	0.69	0	0
Animals	0	0	0.16	0.43	0	0.04	0	0.13	0	0	0	0

(d) Only PVs with: aus

	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	0.83	0	0.53	0	0.42	0	0.26	0.47	0	0	0	0.97
Plants	0	1.23	0	0	0	0.45	0	0.01	0	0	0.17	0
Movement / Direction	0.17	0	0	0	0.99	0	0	1.03	0	0	0	1.47
Machines / Tools	0.96	0.14	0.11	0.15	0	0	0	0.63	0	0	0	0
Light / Darkness	0	0	0.03	0.6	0	0	0	0	0	0.09	0.29	0
Human−Body	0.5	0	0.11	0	0.2	0	0.91	0.54	0	0	0.16	0
Heat / Cold	0	0	0	0.36	0	0.59	0.22	0	0	0	0.04	0
Health / Illness	0	0.03	0.63	0	0.06	0	0.82	0.54	0	0	0.04	0
Games / Sports	0	0	0	0.38	0.06	0	0	0	0	0	0.22	0.4
Forces	0.85	0.79	0.72	0	0.44	0	0	0	0	0	0.02	0.14
Economic−Transaction	0	0	0	0.07	0	0.62	0	0.9	0	1.72	0	0
Cooking / Food	0	0	0	0	0	0.51	0.34	0.09	0	0	0.31	0
Buildings / Construction	0	0.25	0	0	0.64	0.6	0	0	0	1.01	0	0.8
Animals	0	0	0.47	0	0.68	0	0.35	0	0	0.61	0.32	0.15

Figure 6.4: (Literal) source domain → (non-literal) target domain per particle

literal BV sentences and non-literal PV sentences. In the same way, Figure 6.4 presents these shifts based on each of the four particle types, using only the corresponding subset of the collection. The cells in the heat map (=strength) present the results of multiplying the target domain degrees of membership across all PVs with the source-domain degrees of membership with regard to their respective BVs.

To illustrate, almost 92% of all literal sentences with "keimen" *(to sprout)* belong to the source domain of Plants. Hence, this verb has a membership degree of .92 to this source domain. While this is the case, the PV "aufkeimen" *(to bud (fig.))* belongs strongly to the class Desire, according to its non-literal sentences. Hence, we multiply both of the membership degrees and add the result to the current value in cell Plants→Desire. Similarly, "saufen" *(to guzzle)* has a strong membership degree to the source class Animals and its PV "absaufen" *(to drown)* belongs strongly to the class Life/Death adding a strong value to Animals→'Life/Death. Note that a BV-PV combination can add value to multiple cells.

To avoid a bias toward popular classes, we applied positive pointwise mutual information (PPMI) weighting[3]. The results show multiple interesting combinations. Examples of particularly strong combinations are Plants → Time (e.g., *blühen → aufblühen*) and Sound → Communication (e.g., *bellen → anbellen*). Interestingly, some patterns confirm theories from English metaphor detection (Kövecses, 2002). To elaborate, for many of our highly associated pairs, we find examples from Köveces. For example, Human-Relationships is strongly associated with multiple source domains, namely Economic-Transaction (*she invested a lot in that relationship*), Buildings/Construction (*we are as one*), and Heat/Cold *(burning with love)*. Furthermore, our Plants and Economy confirms the general and commonly used mapping social organizations are plants.

In addition, based on the unweighted pairwise association strength between the source and target domains, we computed strength per domain and looked at the most common used domains. We restricted the analysis again to the subset of literal sentences for source domains and non-literal sentences for target domains.

The three source domains with the highest strength are:

1. Human-Body
2. Animals
3. Machines/Tools

[3] We present the plot based on the unweighted association strength in the Appendix (Figure 8.1).

In the same way, the three highest target domains are:

1. Emotion/Feeling
2. Thought
3. Human-Relationships

Again, these findings are highly similar to the assumptions by Köveces. Here, Human-Body, Health/Illness and Animals are assumed to be the three most used source domains. However, our top target domains do not match the Köveces expectations, except for Emotion/Feeling. Köveces expects Emotions, Desire, and Morality to be the three most commonly used sources for target domains.

6.1.3 Summary

We presented a new collection to assess meaning components in German PVs, by relying on a novel strategy to obtain source and target domain characterizations via sentence generation rather than sentence annotation. We confirmed that non-literal usage was correlated with the usage of abstract context words. Finally, our computational model revealed patterns from source to target mappings that are remarkably similar to assumptions from metaphor literature.

6.2 Type-Based Regular Meaning Shifts

In this section, we look at pairs of BV-PV combinations and address the question of regular meaning shifts. According to the case-study by Springorum et al. (2013b), there are some BVs that gain a new (shifted) sense and behave analogous when combined with the same particle. Furthermore, these examples show that such shifts often apply across a semantically coherent set of verbs.

In this chapter, we explore to what extent such regular shifts exist. Furthermore, we build a computational model to classify various cases of meaning shifts/no-shifts. This is done by using a type-based model of analogy.

6.2.1 Introduction

German PVs often trigger meaning shifts of the base verbs (Springorum et al., 2013b; Köper and Schulte im Walde, 2016). More specifically, Springorum et al. (2013b) presented a manual corpus exploration suggesting regular mechanisms in meaning shifts from base verbs to PVs that apply across a semantically coherent set of BVs. For example, the two sound BVs "brummen" (*to hum*) and "donnern" (*to rumble*) both describe a displeasing loud noise. Combining them with the particle *auf*, the PVs *aufbrummen* and *aufdonnern* are near-synonyms in one of their senses, roughly meaning *"to forcefully assign a task"*. In a similar vein, Morgan (1997) used schematic diagrams to illustrate meaning shifts of English complex verbs with the particle *"out"*.

The goal of this work is to provide a computational model of meaning shifts for German PVs. We define our task from the perspective of an analogy, comparing a BV pair with a PV pair, cf. Figure 6.5. A BV–PV model of regular meaning shifts expects:

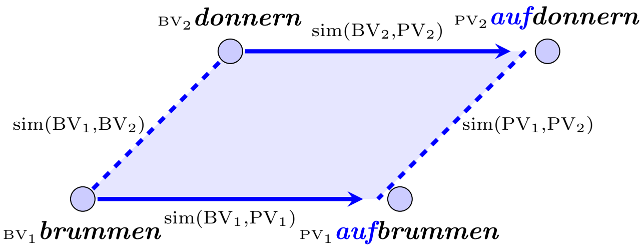


Figure 6.5: Analogy model applied to BV–PV shifts.

1. Semantic coherence *sim(BV$_1$,BV$_2$)* between the two BVs (i.e., overlap in a selected set of semantically salient features)
2. Strong semantic similarity *sim(PV$_1$,PV$_2$)* between the PVs.
3. Low semantic similarity *sim(BV$_i$,PV$_i$)* between the corresponding BV–PV pairs, where the shifts take place.

In a similar vein, a rich tradition on computational work on analogies focuses on finding a relational analogy in multiple choices as required by the Scholastic Aptitude Test (SAT) (Turney, 2006, 2012; Speer et al., 2017). While the SAT questions provide a limited set of possible answers, more recent attention has been spent on open vocabulary tasks of the form *A:B::C:?* (Mikolov et al., 2013d; Levy and Goldberg, 2014a).

The contribution of our analogy model is two-fold: (i) it takes a step forward from hand-selected manual datasets of meaning shifts to larger-scale automatic classification; and (ii) it aims to deepen linguistic insights into complex verb meaning shifts. While we focus on German PVs, we expect our explorations to be applicable to other types of meaning shifts or languages. Most importantly, we show that (a) there are variants of (ir)regular meaning shifts that go beyond what was found in corpus-based explorations; (b) generalization via classification boosts the strengths of salient verb features; and (c) affective features (i.e., abstractness, emotion and sentiment) play the predominant role in similarity models of meaning shifts.

6.2.2 A Collection of BV–PV Pairs

To our knowledge, apart from small-scale case studies (Springorum et al., 2013b), no datasets of human-annotated complex verb meaning shifts are available. Therefore, we collected human judgments for the combinations of BV–PV pairs of the form:

$$BV_1 : PV_1 :: BV_2 : PV_2$$

such as *klappern : abklappern :: klopfen : abklopfen*. We aimed for ≈200 BV–PV pairs per particle type, focusing on the four highly frequent particle types, *ab, an, auf, aus*. The target selection was restricted to $\mathrm{PV}_1/\mathrm{PV}_2$ combinations with identical particles, and where the two PVs were deemed (near-)synonyms according to the German standard dictionary Duden[4] or the German Wiktionary[5], since we were interested in BV–PV analogies with highly semantically similar PVs. In addition, we added interesting cases from the literature. In total, we collected 794 BV–PV pairs questions.[6] The BV–PV pairs were distributed over four lists according to the four particle types, and were annotated by five German native speakers with a background in linguistics. To avoid a sense-specific bias, we provided no contextual information and, therefore, conducted the annotation on the type level. The annotators were asked to classify the pairs according to multiple questions, shown in Figure 6.6. Based on their answers, we were able to assign each BV–PV pair into four categories to distinguish between near and non-analogies, as well as meaning shifts in no/one/both BV–PV pairs:

1. **Comp**: No BV–PV pair has a meaning shift, i.e., both PVs are compositional regarding their BVs and, therefore, all the four verbs are (near-)synonyms. For example: "(ab)feilen::(ab)schleifen" *to grind (off)*. The resulting pairs can be seen as near-analogy (see Turney (2006)).

[4] www.duden.de/suchen/dudenonline.

[5] https://www.wiktionary.org.

[6] The dataset is publicly available at: www.ims.uni-stuttgart.de/data/pv-meaning-shift.

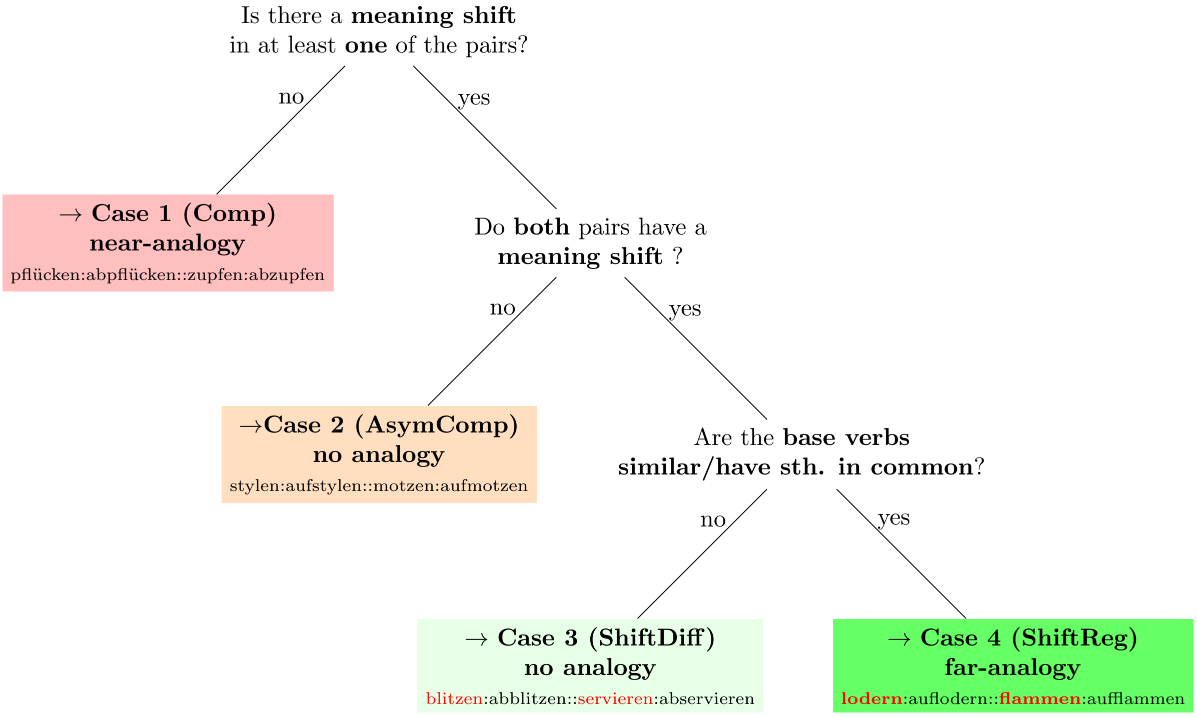

Is there a **meaning shift** in at least **one** of the pairs?
no
yes
→ **Case 1 (Comp)** **near-analogy** pflücken:abpflücken::zupfen:abzupfen
Do **both** pairs have a **meaning shift** ?
no
yes
→**Case 2 (AsymComp)** **no analogy** stylen:aufstylen::motzen:aufmotzen
Are the **base verbs** **similar/have sth. in common**?
no
yes
→ **Case 3 (ShiftDiff)** **no analogy** blitzen:abblitzen::servieren:abservieren
→ **Case 4 (ShiftReg)** **far-analogy** **lodern**:auflodern::**flammen**:aufflammen

Figure 6.6: Tree annotation scheme for PV pairs.

2. **AsymComp**: Only one of the BV–PV pairs undergoes a meaning shift; in this case, the annotators also indicated that pair. For example: “(auf)wühlen::(auf)graben” lit. *to churn::dig (up)*, where “aufwühlen” includes a strong emotion component. In this case, the BV–PV pairs form no analogy.

3. **ShiftDiff**: Both BV–PV pairs show a meaning shift, but the BVs are not semantically similar. For example: “(aus)baden::(aus)bügeln” *to pay for an error* with “baden” *to take a bath* and “bügeln” *to iron*. Note that, according to the definition of Turney (2006), these BV–PV pairs form no analogy.

4. **ShiftReg**: Both BV–PV pairs undergo a meaning shift, and the BVs are semantically similar. For example: “(an)graben::(an)baggern” *to hit on sb.* with both “graben” and “baggern” *to dig*. Pairs, belonging to this class, can be seen as far-analogies.

For practical reasons, we merged the left/right asymmetric cases AsymComp such that the annotated meaning shift was always on the left-hand side (by swapping the asymmetric-right pairs), since these cases represent instances of the same phenomenon, i.e., where just one of the pairs underwent a meaning shift.

Despite a distinction into four categories per instance, we obtained a moderate

Fleiss' κ agreement of 0.43 as the mean across the four particles.

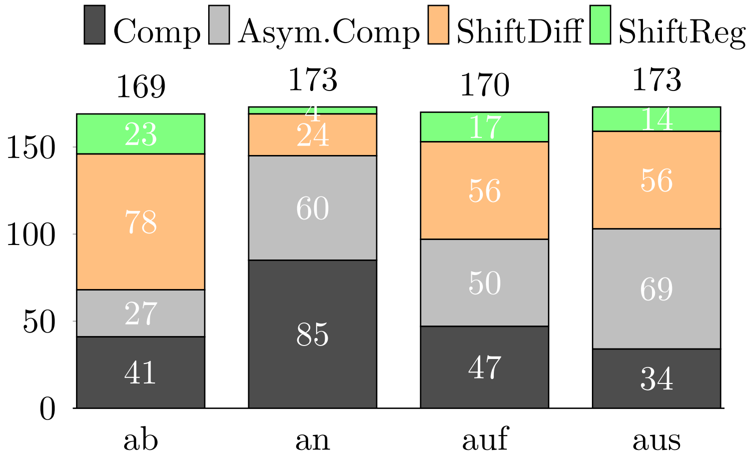


Figure 6.7: Number of majority class instances for four meaning shift categories by particle type.

We transformed the annotations to actual class assignments by removing all the instances from the dataset without a category majority i.e., we only included BV–PV analogy pairs where at least three out of five annotators agreed on the shift category. We assigned the majority decision as the class label. The final collection still contains 685 BV–PV pairs.

The distribution across the four particles and the four categories is illustrated in Figure 6.7; examples are listed in Table 6.3. While meaning shifts have been observed across all four particle types, the analogical case SHIFTREG mentioned in the previous corpus explorations represents the smallest class overall (8.5%). For the particle *an*, cases with two meaning shifts (SHIFTDIFF+SHIFTREG) are especially rare (16.2%).

A manual inspection revealed that etymology and semantic change often led to opaque PVs annotated as SHIFTDIFF; an example is "abkupfern", *to plagiarise*. The origin of this meaning is based on the 18th century engravers who etched replicas of text and images into copper ("Kupfer") plates.

6.2.3 Representations of BV–PV Pairs

The parallelogram in Figure 6.5 illustrates the (dis-) similarities between BVs and PVs that come into play when distinguishing between the four types of (non-)shifts in our dataset: COMP requires all the four sides in the parallelogram to provide strong similarities; SHIFTREG requires the BV_i–BV_j and the PV_i–PV_j sides to provide

Comp	AsymComp	ShiftDiff	ShiftReg
abfeilen::abschleifen	abbauen::abmontieren	abschreiben::abkupfern	abstottern::abrattern
abkuppeln::abhängen	abchecken::abprüfen	abschweifen::abdriften	abrauschen::abzischen
absenden::abschicken	abdampfen::abdunsten	abblitzen::abservieren	abspeisen::abfrühstücken
aneignen::anlernen	anfeuern::anbrennen	ankreiden::anlasten	anheizen::anfeuern
anbrüllen::anschreien	anhängen::anheften	anfechten::angreifen	anwerfen::anschmeißen
anmurren::anknurren	anmachen::anpöbeln	anlachen::anmachen	angraben::anbaggern
auftupfen::auftropfen	aufdrehen::aufzwirbeln	auftreiben::aufspüren	aufwirbeln::aufrühren
auffuttern::aufessen	aufmotzen::aufstylen	aufkreuzen::auftauchen	aufbrummen::aufdonnern
aufritzen::aufschlitzen	aufwühlen::aufgraben	auferlegen::aufbrummen	aufkeimen::aufblühen
aufbaggern::aufbuddeln	ausmalen::ausdenken	ausbaden::ausbügeln	ausdrücken::ausquetschen
ausrupfen::ausjäten	ausposaunen::ausplaudern	ausfeilen::ausbrüten	ausweinen::ausheulen
ausschnaufen::ausatmen	aussaugen::auspumpen	ausstechen::ausbremsen	auskochen::ausbrüten

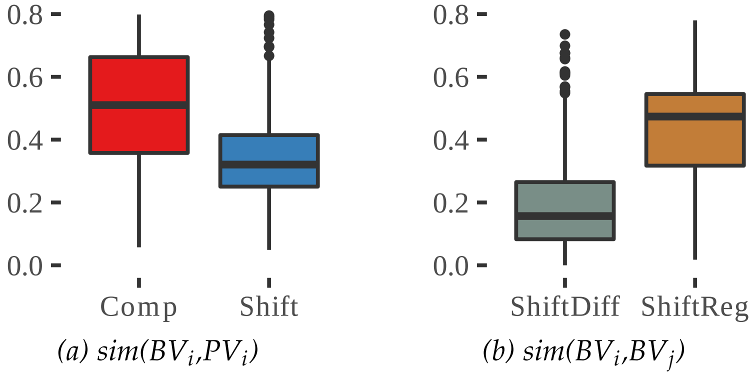
Table 6.3: Example of BV–PV analogies across the four meaning shift categories.

strong similarities, and both BV_i–PV_i sides to provide strong dissimilarities, etc. An obvious option to address the classification of the BV–PV analogies is, thus, by relying on standard cosine scores, when representing the verbs in a DSM. The following paragraphs describe such a basic cosine-similarity model that we used as a baseline, as well as alternative features that we added as potentially salient regarding our task.

Basic Distributional Similarity Model

We created a basic DSM to represent all BVs and all PVs by using a corpus-derived 300-dimensional vector representation. We applied the reconstruction of separated PVs as a preprocessing step. Following our previous experiments, we relied on *DECOW14AX* (Section 3.5) as corpus resource. The verb vectors were obtained by looking at all the context words within a symmetrical window of size 3. We applied positive PPMI feature weighting together with SVD Measuring the cosine similarities between the BVs and PVs, as suggested by Figure 6.5, then, represents our basic distributional similarity model containing four cosine values.

Figure 6.8 looks into the cosine values across combinations of meaning shift categories. Figure 6.8 (a) shows box plots for BV-PV pairs in the two compositional categories vs. the meaning-shifted categories. It illustrates that BV-PV combinations with a meaning shift indeed have lower cosine values between BVs and PVs than BV-PV combinations without meaning shifts. The similarity between BVs is expected to be higher for the regularly shifted cases, where the BVs have something in common, in contrast to the irregular cases. This is also confirmed, (cf. Figure 6.8 (b)).



(a) $sim(BV_i,PV_i)$ (b) $sim(BV_i,BV_j)$

Figure 6.8: Cosine distributions across categories.

Generalization Models

Classes and clusters are powerful techniques to generalize for unseen or infrequent events. Therefore, we extended the basic similarity model by adding class label features for all four involved verbs. We compared three different classifications:

1. We used the 15 verb classes from *GermaNet* (Hamp and Feldweg, 1997; Kunze, 2000). For PVs not covered by *GermaNet*, we used the existing verbs as a seed set and applied a nearest-prototype (centroid) classifier to all other BVs and PVs, with a centroid for each of the 15 classes. Thus we were able to assign class labels to all the verbs in our dataset.

2. For three out of our four particle types (*ab, an, auf*), we found existing manual semantic classifications with 9, 8, and 11 classes, respectively (Lechler and Roßdeutscher, 2009b; Kliche, 2011; Springorum, 2011). To obtain class labels for all verbs, we applied the same nearest-centroid technique as for the *GermaNet* classes. It must be mentioned, that these semantic classes are the same ones used in our previous experiment (Section 5.2.4) and are shown in Appendix 8.

3. We compared the two resource-based methods with an unsupervised k-Means clustering based on verbs' vector representations. Unlike the other methods, k-Means learns the centroids without manually defined seed assignments. We set the number of clusters to $k = 10$, since this granularity was similar to the manual classifications.

Affect Models

According to the empirical study on English metaphors from Mohammad et al. (2016), metaphorical usages are, on average, significantly more emotional than literal usages. Similarly, a BV–PV meaning shift often involves a change in emotion and/or sentiment. For example, while the BV "servieren" (*to serve*) is perceived as rather neutral or slightly positive, the PV "abservieren" (*to dump sb.*) has a clearly negative meaning and correlates with the emotion of *sadness*. On the other hand, the BV "motzen" (*to grumble*) is associated with a negative sentiment and the emotion of *anger*, while its PV "aufmotzen" (*to shine up, soup up*) indicates a positive change.

In a slightly different vein, non-literal word usage often correlates with the degree of abstractness of the word's contexts (Turney et al., 2011; Tsvetkov et al., 2014; Köper and Schulte im Walde, 2016). For example, the PV "abschminken" with the BV "schminken" (*to put on make-up*) has a literal, very concrete meaning (to remove make-up) and also a shifted, very abstract non-literal meaning (*to forget about something*).

We enriched the basic similarity model by integrating affective information from manually created lexicons. Since affective datasets are typically small-scale and mostly exist for English, we applied a cross-lingual approach to propagate ratings to German. The procedure, to learn these norms, is explained in Section 3.4.2.

The procedure was applied to a range of affective norm datasets in isolation: The *NRC Hashtag Emotion Lexicon* (Mohammad and Kiritchenko, 2015) contains emotional ratings for 17000 words; we used *anger*, *disgust*, *fear*, *joy*, and *sadness*. Warriner et al. (2013) collected 14000 ratings for *valence* and *arousal*. For *concreteness*, we relied on the 40000 ratings from Brysbaert et al. (2014). Finally, we used the 10000 ratings for *happiness* from Dodds et al. (2011). In total, we obtained nine affective values for 2.2 million words.[7]

We added the affective features to our basic similarity model by first looking up the 9-dimensional affect vectors for all our four verbs involved in an analogy and, then, calculating for each of the four similarities in the analogy parallelogram (Figure 6.5) the element-wise differences between the nine affective dimensions of the respective two verbs, resulting in $4 \times 9 = 36$ extra vector dimensions.

Recent work from Lapesa et al. (2017) shows that such norms can be crucially improved if the word's context is relied on instead of the words direct value. Therefore,

[7] These ratings are also available at `www.ims.uni-stuttgart.de/data/pv-meaning-shift`.

in addition to looking at the verbs' affective values, we also looked at the affect values of the respective context words: For each verb, we created a second 9-dimensional vector with average affective values across the 500[8] most associated context words, according to PPMI. With respect to the four verbs in the analogy, this resulted in another $4 \times 9 = 36$ extra vector dimensions.

We further added affect information restricted to the common context words of the involved verbs (red and blue intersections in Figure 6.9): For each intersection of the two BVs and the two PVs as well as the two BV–PV combinations, we learned another 9-dimensional emotional centroid, now only based on the shared context words, and we provided the element-wise differences between the two centroids as a feature. By focusing on the intersections, we strengthened words that both verbs have in common, focusing on a particular sense.

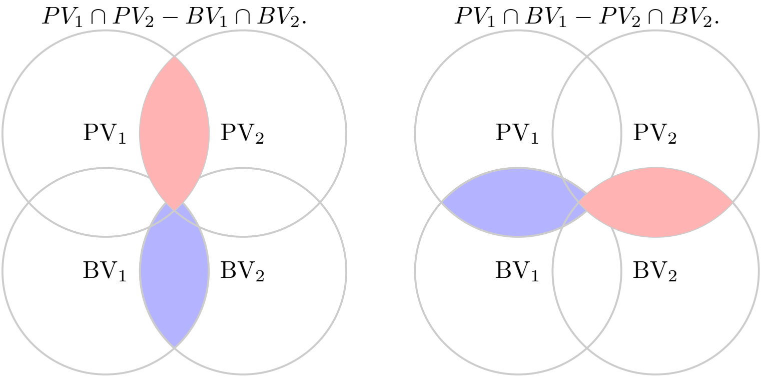

$PV_1 \cap PV_2 - BV_1 \cap BV_2.$
PV$_1$
PV$_2$
BV$_1$
BV$_2$
$PV_1 \cap BV_1 - PV_2 \cap BV_2.$
PV$_1$
PV$_2$
BV$_1$
BV$_2$

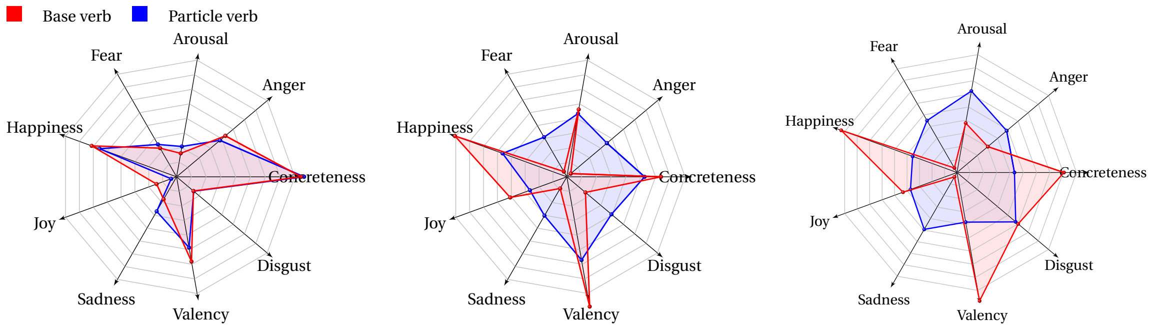
Figure 6.9: Venn diagrams with intersections.

6.2.4 Experiments on BV–PV Pairs

Two classification scenarios were implemented: a four-class distinction between our four shift categories (*4-Classes*), and a binary distinction between cases where both BV–PV pairs include a meaning shift (ShiftDiff+ShiftReg) vs. BV–PV pairs that include cases of compositionality (Comp+Asym.Comp). We refer to this setup with *Shift-vs-Comp*.

We applied a supervised classification setting based on SVMs with an RBF kernel (Chang and Lin, 2011), using 10-fold cross-validation. Next to the similarity, generalization and affect features, we provided the particle type as a feature in all the

[8] Experiments with other values within $[50, 1000]$ showed the same behavior



COMP: (ab)montieren
(to mount → to dismount)

SHIFT: (ab)frühstücken
(to have breakfast → to fob sb. off)

SHIFT: (ab)servieren
(to serve → to dump sb.)

Figure 6.10: Changes in affect and emotion for one compositional and two shifted BV–PV pairs. The affect/emotion values are based on the top associated context words according to PPMI.

settings. Table 6.4 reports the results across the feature sets. As an evaluation metric, we report accuracy and a macro-average (equally-weighted, Section 3.2.2) f-score (F_1) over all classes.

	4-Classes		Shift-vs-Comp	
	Acc	F_1	Acc	F_1
Majority baseline	31.24	12	60.29	38
Basic Sim	40.73	32	65.10	60
Sim+GermaNet	43.36	34	67.15	59
Sim+ManClass	45.55	36	69.05	62
Sim+k-Means	52.99	37	70.51	66
Affect (full)	57.08	44	76.49	74
Affect only verbs	47.73	37	69.05	65
Affect only context	**58.39**	**45**	**78.54**	**77**
Affect only context w/o intersec.	52.55	.40	72.40	68
Affect only context-only intersec.	54.60	.42	73.72	72
Combination	56.20	44	77.08	75

Table 6.4: Results for 4- and 2-class distinctions, reporting accuracy and macro-F_1.

All the models performed significantly[9] better than the majority baseline. In addition, the full and the context-only affective models performed significantly better than the similarity models with and without generalization, even though the unsupervised k-Means clustering improves the basic similarity model significantly

[9] Significance relies on χ^2 with $p < 0.001$.

(*Sim+k-Means*). Finally, the context-only affective model outperformed the verb-only affective model. Here, in particular, the subset of the affective features, derived from the intersections (context-only intersec.) provide useful information. Interestingly, a combination of all the features (*Combination*) did not perform better than the context-only affective model in isolation.

A leave-one-out classification using the best classifier *Affect-only context* as starting point revealed that most performance (accuracy) is lost when removing the emotion of *fear* (-2.77), followed by the emotions of *joy* (-1.46) and *arousal* (-0.88). In contrast, features related to *disgust* showed no impact on the overall performance.

Figure 6.10 illustrates that we can spot changes in affect and emotion even on the verb level: For three BV–PV verb pairs with particle *ab*, it plots the nine affective and emotion ratings for both verbs, after rescaling to an interval of $[0, 10]$. In the compositional case (a) the PV is highly similar to the BV in all dimensions, creating roughly the same shape as the BV. In the shift cases (b) and (c), the PVs are less concrete and evoke less *happiness* and *joy* than the BVs, while they evoke more *fear*, *anger* and *sadness* in comparison to their BVs.

6.2.5 Summary

This work presented a computational model of meaning shifts for German PVs. Relying on a novel dataset, we found that shifts were observed across all our four particle types, but the analogical case mentioned in previous corpus explorations only represented the smallest class overall (8.5%). SVM models successfully distinguished between shift categories, with verb classes boosting standard cosine similarity performance, and affective context features representing the most salient indicators.

6.3 Chapter Summary

In this chapter, we studied patterns and regular meaning shifts of PVs. We presented a collection and statistical analysis of BV and PV sentences, annotated for non-literalness and domain membership. Next, we created a BV-PV analogy collection and a computational model to distinguish various meaning shifts.

Here, our domain collection presented a novel way to assess changes in meaning. By measuring association strength between domains, we could study repeating patterns from one domain to another. The collection allowed us to extract popular source and target domains. Furthermore, we could zoom into particle-specific contributions as well as changes from literal BV usage to non-literal PV usage.

Our analogy setup showed a large amount of BV-PV meaning shifts. However, cases with pairs of coherent BVs representing a regular meaning shift were observed less frequently. In addition, we showed that a supervised computational model is able to classify various cases of meaning shifts. Such a model could be improved by including verb generalizations via clustering. Besides, a model that additionally accounts for changes in affect and sentiment by relying on automatically created norms showed best performance.

7 Conclusion

This chapter summarizes the main findings and results of the thesis, and outlines some ideas for future work to address the limitations of the thesis.

7.1 Conclusion

This thesis presented various computational experiments on the challenging class of German Particle verbs.

Overall, the approaches cover a wide selection of challenging phenomena, including the prediction and modeling of compositionality, senses and non-literal language. In addition, this thesis addressed PVs at a large-scale and across multiple particles. For that reason, the presented research can be seen as rather broad, and consequently, some aspects lack in-depth analysis.

The contributions in this thesis cover practical and theoretical aspects. On the practical side, we developed and implemented specific computational models that are potentially interesting for downstream application, e.g., to detect non-literal PV usage with high accuracy. On the theoretical side, we were able to extract salient information e.g., the importance of affective norms. In addition, our models provide new insights into patterns from source to target domain mappings. Another essential point is that we confirm the difficulty of the phenomena.

More importantly, there were some striking findings across the different research directions. Across tasks, we observed that semantically similar BVs exhibit similar behavior when combined with a particle. This pattern was confirmed for predicting compositionality, as well as non-literal language detection. The compositionality experiment, which aimed at predicting a PV vector representation, achieved superior performance when applying training-space restrictions. Thus, learning from a small but semantically similar set of BVs provided better predictions than learning from a larger but potentially semantically less related set of verbs. In the same way, we observed increased performance when PVs with identical particles and semantically similar BVs predicted each others literal and non-literal usage.

Furthermore, multiple experiments confirmed that a plain textual distributional model is often not enough. Adding external information provides complementary and useful information. Such information can be integrated via visual representations of images or via human-created external affective or emotion norms. The usefulness of such norms in computational models is a central finding in this thesis. These norms were important features for the detection of non-literal language (abst./conc. norms) as well as the most salient feature in our analogy classification setup. In a similar vein, our findings suggest that distributional models that account for different senses perform better than traditional models. Moreover, we showed that BV and PV meanings are better captured in distributional representations that account for syntactically separated verb occurrences.

While PVs represent an exceptional phenomenon, we were able to utilize PV-specific features. An interesting example is the distributional-fit feature that we deploy for the enhanced detection of non-literal PV usage. This feature simply computes the similarity between context and a PV's corresponding BV. Although the feature does not seem very sophisticated at first glance, it combines compositionality prediction and non-literal language detection. Furthermore, it makes use of information that is only available for such verbs. In the same way, we observed that providing a particle as a feature is often beneficial.

Our novel collection and analysis based on source and target domains revealed interesting patterns of commonly used domain changes from literal to non-literal language. Complementary to this, our analogy setup found only a few instances where two BVs, with obvious common properties, undergo the same meaning shift. Consequently, the phenomenon of regular meaning shifts seems less widespread than we previously assumed.

That aside, the computational models in this thesis can be divided into ones where a verb is studied in context or running text (token-based) and approaches that treat PV meaning across all contexts (type-based). In this thesis, we make use of both perspectives. Currently, a lot of interest in distributional semantics, and particularly with respect to embedding learning, is spent entirely on type-based techniques. One reason for this is that token-based approaches require expensive manual annotations.

In hindsight, and with respect to our models, the token-based approaches gave us more insights into the phenomena and were worth the annotation effort. In brief, computational models working on token-based usage of PV were more transparent and allowed a better interpretation of the models and results. To illustrate, error and feature analysis for non-literal language classification is comparably easy when one can look at actual sentences and their properties. Similarly, our sentence and domain collection represents a transparent pipeline where one can retrace why a source-target domain combination is associated according to the underlying verbs and their specific usage in context. However, it is comparably difficult to get insights and to draw conclusions from type-based approaches. Furthermore, looking at nearest neighbors, which is not possible for thousands of ambiguous words and various models, one cannot assess which senses a type-based multi-sense representation model captures. In a similar vein, one can hardly zoom into certain properties when measuring compositionality via Spearman's ρ correlation in a sorting task setup. Consequently, there is often no simple answer for seemingly easy questions such as "which are the PVs that are better represented with visual information?".

An analogous conclusions can be drawn with respect to experimental setups and their evaluation techniques. Computational models are usually concerned with obtaining high evaluation metrics. While this is the case, differences in performance, as measured by Spearman's ρ or Fuzzy-BCubed F_1, are hard to grasp and there is no intuitive interpretation in a performance gain of 5 points. Complementary to this, a classification setup that reports multiple interpretable measures and results in comparison to a simple majority class baseline provides a better understanding of a computational model.

7.2 Future Work

We will now discuss the shortcomings and limitations of the research presented in this thesis. Furthermore, we present ideas for future directions.

PV Detection: First, in this thesis, we relied on the output of a parser to detect syntactically separated PVs. During the work on PV statistics (Section 3.5.2), manual inspections showed a considerable amount of parsing errors and false reconstructions. Considering that the correct detection of a PV represents the first step in any processing pipeline, we must assume that the detection-step has a major impact on the quality of any computational approach to PVs. Despite, the current approach is not able to distinguish PVs from non-separable prefix verbs. To illustrate, verbs such as "umschreiben" or "durchlaufen" have a separable (PV) and a non-seperable prefix verb reading (see Section 2.1).

Hence, improved PV-detection represents one fundamental and necessary direction for future work.

Token-Based WSD: A second shortcoming in this thesis is the lack of token-based word sense disambiguation. With an eye on downstream applications, a sophisticated model to perform WSD for particle verbs would be beneficial. Here, it could be interesting to establish whether features similar to those used in our literal vs. non-literal setup (Section 5.1), can be applied. Since there is currently no large corpus annotated for PV senses in running text, we were not able to address token-level sense disambiguation. Therefore this problem remains open for future work on PVs.

Semantic Classification of Particle Types: Until now, related work, and this thesis, have conduct semantic classification on the verb level. Complementary to a classification at verb-level, there is no work that aimed at classifying the particles automatically. Such a model would allow novel insights into particle usage e.g., if a certain particle is mostly used directionally vs. aspectually. Here, the rich theoretical literature could provide useful resources. Hence, particle meaning could be modeled in the same way as our past work on automatic semantic classification for German prepositions (Köper and Schulte im Walde, 2016).

Improved Methods to Assess Compositionality: Furthermore, the evaluation of distributional representations is still an open issue. Although this thesis is not concerned with the evaluation of representations, we evaluate the predictions of compositionality by applying a commonly used word similarity schema. Here, performance is expressed by the correlation of the distance between vectors and human judgments of similarity. It has been shown that this kind of evaluation has some flaws (Faruqui et al., 2016). As an alternative, compositionality can also be evaluated via contextual and, therefore, sense-specific judgments, similar to the Stanford's Contextual Word Similarities dataset (Huang et al., 2012). For instance, one could ask human judges to which degree a BV could fit into a given prototypical PV context. Another promising alternative is extrinsic evaluation, where a model's performance is measured via downstream tasks. Such an application can be created for PVs as well, a model that is able to find and mark separated and opaque PVs could provide this information to enhance a machine translation model. In this way, a model's ability to predict compositionality would be evaluated within another task.

Diachronic Representations and Semantic Change: Although we modeled senses and non-literal languages, we did not address the phenomenon of semantic change. A corpus reflects meaning only with respect to a certain zeitgeist; hence, word meaning is usually modeled as a static phenomenon. Thus, taking the temporal dimension into account is definitely an alternative and interesting research direction. A growing interest in research combines methods from distributional semantics with time annotated corpora to study and model the diachronic nature of words (Hamilton et al., 2016; Schlechtweg et al., 2017). Such models could provide a better understanding of PVs and even model their ability to lose and gain senses over time.

Modeling PV Neologisms and Semantic Plausibility: Another interesting future direction is the modeling of PV neologisms in the context of semantic plausibility. Here the question is "can a computational model predict the meaning of a new and potentially unseen PV construction?" and furthermore, "is a certain particle-BV combination plausible?". For example, it is desirable and challenging to build a computational model that assigns low semantic plausibility for a combination between a verb like "schlafen" (*to sleep*) and a directional particle, such as the directional-meaning of "an". On the other hand a combination between "schlafen" and the partitive meaning of "an" can be semantically more plausible. In a similar vein, Wang et al. (2018) show that distributional models without world knowledge struggle with plausibility tasks. The task could be defined as a classification, ranking or even a regression setup. Furthermore, it could be addressed by using compositional distributional approaches (similar to our approach in Section 4.3). However, the focus here would be on low-frequency or non-existing constructions. Gutierrez et al. (2016) recently showed that such compositional models can be used to distinguish between literal vs. non-literal usage hence, they can even account for various possible senses.

Using Novel Techniques: Finally, NLP and more generally all the subfields of AI are fast developing research areas that are enjoying increasing popularity. Currently, there is a lot of interest on neural network or deep learning techniques. It remains an open question if these techniques open entirely new research directions. For computational approaches, new techniques or datasets can be beneficial across fields and lead to new approaches for other research questions.

To illustrate, in this thesis, we make use of multiple recent advances, such as the increasing popularity of embedding learning. Furthermore, our multi-modal space relies on the visual representation of a CNN. Consequently, better models for object-recognition can lead to enhanced multi-modal representations.

Improvements are also made with respect to new tasks, such as image captioning. While the focus here is predominantly on concrete nouns and the English language, such visual applications are potentially also of interest to model the spatial or directional contribution of particles.

Looking at recent publications, there has been consistent work on learning better compositional and non-compositional phrase embeddings or properties that can be used to detect non-compositionality (Yu and Dredze, 2015; Hashimoto and Tsuruoka, 2016; Gong et al., 2017).

In a similar vein, multi-sense representation learning is still an active field. Here, it remains an open question whether sense-specific representation learning is actually useful or not. While our results confirm the need for sense-specific embeddings, other work showed superior performance of single-sense embeddings in downstream applications (Li and Jurafsky, 2015)[1]. With an eye on the impressive results on cross-lingual NMT (Johnson et al., 2016), it seems that the current generation of models for downstream applications are powerful enough to disambiguate words in context without requiring previously divided multi-sense representations.

[1] Li and Jurafsky (2015) find that multi-sense embeddings do improve performance on some tasks but they also find that single-sense embeddings with higher dimensionality gain the same improvements.

Bibliography

Aedmaa, E. (2014). Statistical Methods for Estonian Particle Verb Extraction from Text Corpus. In *Proceedings of the ESSLLI 2014 Workshop: Computational, Cognitive, and Linguistic Approaches to the Analysis of Complex Words and Collocations*, pages 17–22.

Aedmaa, E. (2017). Exploring Compositionality of Estonian Particle Verbs. In *Proceedings of the ESSLLI 2017 Student Session*, pages 197–208.

Aedmaa, E., Köper, M., and Schulte im Walde, S. (2018). Combining Abstractness and Language-specific Theoretical Indicators for Detecting Non-Literal Usage of Estonian Particle Verbs. In *Proceedings of the NAACL Student Research Workshop*, pages 117–218.

Agirre, E. and Soroa, A. (2007). SemEval-2007 Task 02: Evaluating Word Sense Induction and Discrimination Systems. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 7–12, Prague, Czech Republic. Association for Computational Linguistics.

Akmajian, A., Demer, R., Farmer, A., and Harnish, R. (2001). *Linguistics: An Introduction to Language and Communication*. CogNet.

Altarriba, J., Bauer, L., and Benvenuto, C. (1999). Concreteness, Context Availability, and Imageability Ratings and Word Associations for Abstract, Concrete, and Emotion Words. *Behavior Research Methods*, 31(4):578–602.

Alverson, H. (1994). *Semantics and experience: universal metaphors of time in English, Mandarin, Hindi, and Sesotho*. Parallax (Baltimore, Md.). Johns Hopkins University Press.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. *Information Retrieval*, 12(4):461–486.

Apidianaki, M. (2010). An Algorithm for Cross-lingual Sense Clustering tested in a MT Evaluation Setting. In *Proceedings of the 7th International Workshop on Spoken Language Translation*, pages 219–226, Paris, France.

Arthur, D. and Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Bagga, A. and Baldwin, B. (1998). Entity-based Cross-document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pages 79–85, Montréal, Canada.

Baldwin, T. (2005). Deep Lexical Acquisition of Verb–Particle Constructions. *Computer Speech and Language*, 19:398–414.

Baldwin, T., Bannard, C., Tanaka, T., and Widdows, D. (2003). An Empirical Model of Multiword Expression Decomposability. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 89–96, Sapporo, Japan.

Baldwin, T. and Villavicencio, A. (2002). Extracting the Unextractable: A Case Study on Verb Particles. In *Proceedings of the Sixth Conference on Computational Natural Language Learning*, pages 98–104, Taipei, Taiwan.

Bannard, C. (2005). Learning about the Meaning of Verb–Particle Constructions from Corpora. *Computer Speech and Language*, 19:467–478.

Bannard, C., Baldwin, T., and Lascarides, A. (2003). A Statistical Approach to the Semantics of Verb-Particles. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 65–72, Sapporo, Japan.

Baroni, M., Bernardi, R., and Zamparelli, R. (2014a). Frege in Space: A Program for Compositional Distributional Semantics. *Linguistic Issues in Language Technologies*, 9(6):5–110.

Baroni, M., Dinu, G., and Kruszewski, G. (2014b). Don't count, predict! A Systematic Comparison of Context-counting and Context-predicting Semantic Vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247, Baltimore, MD.

Baroni, M. and Kilgarriff, A. (2006). Large Linguistically-processed Web Corpora for Multiple Languages. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.

Barsalou, L. W. and Wiemer-Hastings, K. (2005). Situating Abstract Concepts. *Grounding cognition: The role of perception and action in memory, language, and thought*, pages 129–163.

Batinić, D. and Schmidt, T. (2017). Reconstruction of Separable Particle Verbs in a Corpus of Spoken German. In *International Conference of the German Society for Computational Linguistics and Language Technology*, pages 3–10. Springer.

Bebis, G. and Georgiopoulos, M. (1994). Optimal feed-forward neural network architectures. *IEEE Potentials*, pages 27–31.

Beigman Klebanov, B., Leong, C. W., and Flor, M. (2015). Supervised Word-level Metaphor Detection: Experiments with Concreteness and Reweighting of Examples. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 11–20, Denver, Colorado.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

Bhatia, A., Teng, C. M., and Allen, J. (2017). Compositionality in verb-particle constructions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 139–148, Valencia, Spain.

Biber, D., Johansson, S., Leech, G., Conrad, S., and Finegan, E. (1999). Grammar of spoken and written english. *Edimburgh: Pearson Education Limited*.

Biemann, C. (2006). Chinese whispers: An Efficient Graph Clustering Algorithm and Its Application to Natural Language Processing Problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 73–80, Stroudsburg, PA, USA.

Birke, J. and Sarkar, A. (2006). A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language. In *Proceedings of the 11th Conference of the European Chapter of the ACL*, pages 329–336, Trento, Italy.

Birke, J. and Sarkar, A. (2007). Active Learning for the Identification of Nonliteral Language. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 21–28, Rochester, NY.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Biskup, Petr, P. M. S. L. C. (2011). German Particle and Prefix Verbs at the Syntax-Phonology Interface. *Leuvense Bijdragen - Leuven Contributions in Linguistics and Philology*, 97:106–135.

Black, M. (1962). *Models and Metaphors*. Ithaca: Cornell University Press.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blitzer, J., Weinberger, K. Q., Saul, L. K., and Pereira, O. C. N. (2004). Hierarchical Distributed Representations for Statistical Language Modeling. In *In Advances in Neural Information Processing Systems 17*. MIT Press.

Bohnet, B. (2010). Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Booij, G. (1990). The Boundary between Morphology and Syntax: Separable Complex Verbs in Dutch. 3:45–63.

Bott, S., Khvtisavrishvili, N., Kisselew, M., and Schulte im Walde, S. (2016). G$_h$ost-PV: A Representative Gold Standard of German Particle Verbs. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon*, pages 125–133, Osaka, Japan.

Bott, S. and Schulte im Walde, S. (2014a). Modelling Regular Subcategorization Changes in German Particle Verbs. In *Proceedings of the 1st Workshop on Computational Approaches to Compound Analysis*, pages 1–10, Dublin, Ireland.

Bott, S. and Schulte im Walde, S. (2014b). Optimizing a Distributional Semantic Model for the Prediction of German Particle Verb Compositionality. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 509–516, Reykjavik, Iceland.

Bott, S. and Schulte im Walde, S. (2015). Exploiting Fine-grained Syntactic Transfer Features to Predict the Compositionality of German Particle Verbs. In *Proceedings of the 11th Conference on Computational Semantics*, pages 34–39, London, UK.

Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: the Self-Assessment Manikin and the Semantic Differential. *J Behav Ther Exp Psychiatry*, 25(1):49–59.

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014). Concreteness Ratings for 40 Thousand generally known English Word Lemmas. *Behavior Research Methods*, 64:904–911.

Bullinaria, J. A. and Levy, J. P. (2007). Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study. *Behavior Research Methods*, 39(3):510–526.

C. Dunn, J. (1973). A fuzzy relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. 3:32–57.

Carroll, J. and Fang, A. C. (2004). The Automatic Acquisition of Verb Subcategorisations and their Impact on the Performance of an HPSG Parser. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 107–114, Sanya City, China.

Carroll, J., Minnen, G., and Briscoe, T. (1998). Can Subcategorisation Probabilities

Help a Statistical Parser? In *Proceedings of the 6th ACL/SIGDAT Workshop on Very Large Corpora*, pages 118–126, Montréal, Canada.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chierchia, G. and McConnell-Ginet, S. (2000). *Meaning and Grammar (2Nd Ed.): An Introduction to Semantics*. MIT Press, Cambridge, MA, USA.

Church, K. W. and Hanks, P. (1989). Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, Canada.

Clark, S. (2015). *Vector Space Models of Lexical Meaning*, pages 493–522. John Wiley & Sons, Ltd.

Collobert, R. and Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167.

Constable, J. and Curran, J. (2009). Integrating verb-particle constructions into ccg parsing. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 114–118.

Cook, P. and Stevenson, S. (2006). Classifying Particle Semantics in English Verb-Particle Constructions. In *Proceedings of the ACL/COLING Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 45–53, Sydney, Australia.

Copestake, A. and Briscoe, T. (1995). Regular Polysemy and Semi-Productive Sense Extension. *Journal of Semantics*, 12:15–67.

Dancygier, B. and Sweetser, E. (2014). *Figurative Language*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Daumé III, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263. Association for Computational Linguistics.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Dehé, N. (2015). Particle Verbs in Germanic. In Müller, P. O., Ohnheiser, I., Olsen, S., and Rainer, F., editors, *Word Formation. An International Handbook of the Languages of Europe*, pages 611–626. Berlin: De Gruyter.

Dehé, N., Jackendoff, R., McIntyre, A., and Urban, S. (2002). *Verb-Particle Explorations*, volume 1. Walter de Gruyter.

den Dikken, M. (1995). *Particles: On the Syntax of Verb-particle, Triadic, and Causative Constructions*. Oxford studies in comparative syntax. Oxford University Press.

Depraetere, I. and Salkie, R. (2017). *Semantics and Pragmatics: Drawing a Line*. Logic, Argumentation & Reasoning. Springer International Publishing.

Dewell, R. (2011). *The Meaning of Particle/prefix Constructions in German*. Human cognitive processing. John Benjamins Publishing Company.

Dinh, E.-L. D. and Gurevych, I. (2016). Token-Level Metaphor Detection using Neural Networks. In *Proceedings of The Fourth Workshop on Metaphor in NLP*, pages 28–33, San Diego, CA, USA.

Dinu, G., Lazaridou, A., and Baroni, M. (2015). Improving Zero-shot Learning by Mitigating the Hubness Problem. In *Proceedings of the International Conference on Learning Representations, Workshop Track*, San Diego, CA, USA.

Dodds, P. S., Harris, K. D., Kloumann, I. M., Bliss, C. A., and Danforth, C. M. (2011). Temporal Patterns of Happiness and Information in a Global Social Network: Hedonometrics and Twitter. *PLOS ONE*, 6(12):1–26.

Dorr, B. J. and Jones, D. (1996). Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 322–327, Copenhagen, Denmark.

Duan, K.-B. and Keerthi, S. S. (2005). Which Is the Best Multiclass SVM Method? An Empirical Study. In Oza, N. C., Polikar, R., Kittler, J., and Roli, F., editors, *Multiple Classifier Systems*, pages 278–285, Berlin, Heidelberg. Springer Berlin Heidelberg.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Dunn, J. (2013). What Metaphor Identification Systems can tell us About Metaphor-in-language. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 1–10, Atlanta, Georgia.

Eichinger, L. M. (2000). *Deutsche Wortbildung: eine Einführung*. Gunter Narr Verlag.

Eilola, T. and Havelka, J. (2010). Affective Norms for 210 British English and Finnish Nouns. *Behavior Research Methods*, 42(1):134–140.

Emonds, J. (1972). Evidence that indirect object movement is a structure-preserving rule. *Foundations of language*, pages 546–561.

Evert, S. (2005). *The Statistics of Word Co-Occurrences: Word Pairs and Collocations*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Evert, S. and Krenn, B. (2001). Methods for the Qualitative Evaluation of Lexical Association Measures. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 188–195, Toulouse, France.

Faaß, G. and Eckart, K. (2013). SdeWaC – A Corpus of Parsable Sentences from the Web. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*, pages 61–68, Darmstadt, Germany.

Faaß, G., Heid, U., and Schmid, H. (2010). Design and Application of a Gold Standard for Morphological Analysis: SMOR in Validation. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 803–810, Valletta, Malta.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proc. of the 1st Workshop on Evaluating Vector Space Representations for NLP*.

Felfe, M. (2012). *Das System der Partikelverben mit „an ": Eine konstruktionsgrammatische Untersuchung*, volume 12. Walter de Gruyter.

Fellbaum, C., Grabowski, J., and Landes, S. (1998). Performance and Confidence in a Semantic Annotation Task. In Fellbaum, C., editor, *WordNet – An Electronic Lexical Database*, Language, Speech, and Communication, chapter 9, pages 217–237. MIT Press, Cambridge, MA.

Filip, H. (2012). Lexical aspect. In Binnick, R. I., editor, *The Oxford Handbook of Tense and Aspect*, pages 721–751. Oxford University Press, Oxford.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and 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, 1930–55. In Palmer, F. R., editor, *Selected Papers of J.R. Firth 1952–59*, Longman's Linguistics Library, pages 168–205. Longmans.

Firth, R. (1953). The Technique of Semantics. 34:36 – 73.

Fleischer, W. and Barz, I. (2012). *Wortbildung der deutschen Gegenwartssprache*. de Gruyter, Berlin.

Forceville, C. (1994). Pictorial Metaphor in Advertisements. 9:1–29.

Frassinelli, D., Abrosimova, A., Springorum, S., and im Walde, S. S. (2017). Meaning (Mis-)Match in the Directionality of German Particle Verbs. Poster at the 30th Annual CUNY Conference on Human Sentence Processing.

Fritzinger, F. (2010). Using Parallel Text for the Extraction of German Multiword Expressions. *Lexis. Journal in English Lexicology*, (4).

Gardner, D. and Davies, M. (2007). Pointing out Frequent Phrasal Verbs: A Corpus-based Analysis. *TESOL quarterly*, 41(2):339–359.

Gedigian, M., Bryant, J., Narayanan, S., and Ciric, B. (2006). Catching Metaphors. In *Proceedings of the 3rd Workshop on Scalable Natural Language Understanding*, pages 41–48, New York City, NY.

Geeraerts, D. (1997). Diachronic Prototype Semantics. *A Contribution to Historical Lexicology*.

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7:155–170.

Gentner, D., F. Bowdle, B., Wolff, P., and Boronat, C. (2001). Metaphor Is Like Analogy. *The analogical mind: Perspectives from cognitive science*, pages 199–253.

Gerdes, J. (2012). *Partikelverben im produktiven Gebrauch. Eine Korpusuntersuchung verbaler Bildungsschemata in Pressetexten.* PhD thesis, Universität Trier.

Gladkova, A., Drozd, A., and Matsuoka, S. (2016). Analogy-based Detection of Morphological and Semantic Relations with Word Embeddings: what works and what doesn't. In *Proceedings of the Student Research Workshop of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8–15, San Diego, California, USA.

Glenberg, A. M. and Kaschak, M. P. (2002). Grounding Language in Action. *Psychonomic Bulletin and Review*, 9(3):558–565.

Goldberg, Y. and Hirst, G. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Goldberg, Y. and Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s Negative-Sampling word-embedding Method. *arXiv preprint arXiv:1402.3722*.

Gong, H., Bhat, S., and Viswanath, P. (2017). Geometry of Compositionality. In *AAAI*, pages 3202–3208.

Grefenstette, G. (1994a). Corpus-derived First, Second and Third-order Word Affinities. In *In Proceedings of EURALEX*, pages 279–290.

Grefenstette, G. (1994b). *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA.

Groos, A. (1989). Particle-Verbs and Adjunction. *Linguistics in the Netherlands*, 1989:51–60.

Gupta, A., Boleda, G., Baroni, M., and Padó, S. (2015). Distributional Vectors Encode

Referential Attributes. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal.

Gurevych, I. (2005). Using the Structure of a Conceptual Network in Computing Semantic Relatedness. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 767–778, Berlin, Heidelberg. Springer-Verlag.

Gutierrez, E. D., Shutova, E., Marghetis, T., and Bergen, B. (2016). Literal and metaphorical senses in compositional distributional semantic models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 183–193.

Haagsma, H. and Bjerva, J. (2016). Detecting Novel Metaphor using Selectional Preference Information. In *Proceedings of The Fourth Workshop on Metaphor in NLP*, pages 10–17, San Diego, CA, USA.

Hamerly, G. and Elkan, C. (2002). Alternatives to the K-means Algorithm That Find Better Clusterings. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 600–607, New York, NY, USA. ACM.

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1501, Berlin, Germany.

Hamp, B. and Feldweg, H. (1997). GermaNet – A Lexical-Semantic Net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

Harm, V. (2000). *Regularitäten des semantischen Wandels bei Wahrnehmungsverben des Deutschen*. Number 110. Franz Steiner Verlag.

Harnad, S. (1990). The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hartmann, S. (2008). Einfluss syntaktischer und semantischer Subkategorisierung auf die Kompositionalität von Partikelverben. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Haselbach, B. (2011). Deconstructing the Meaning of the German Temporal Verb Particle *'nach'* at the Syntax-Semantics Interface. In *Proceedings of Generative Grammar in Geneva*, pages 71–92, Geneva, Switzerland.

Hashimoto, K. and Tsuruoka, Y. (2016). Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings. *arXiv preprint arXiv:1603.06067*.

Heintz, I., Gabbard, R., Srivastava, M., Barner, D., Black, D., Friedman, M., and Weischedel, R. (2013). Automatic Extraction of Linguistic Metaphors with LDA Topic Modeling. In *Proceedings of the First Workshop on Metaphor in NLP*, pages 58–66, Atlanta, Georgia.

Helbig, G. and Buscha, J. (1998). *Deutsche Grammatik*. Langenscheidt – Verlag Enzyklopädie, 18th edition.

Henzen, W. (1965). Deutsche wortbildung.-3., ergänzte auflage. *Tubingen: Niemeyer*.

Hill, F., Cho, K., Korhonen, A., and Bengio, Y. (2016). Learning to Understand Phrases by Embedding the Dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30.

Hill, F. and Korhonen, A. (2014). Learning abstract concept embeddings from multimodal data: Since you probably can't see what i mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265, Doha, Qatar. Association for Computational Linguistics.

Hill, F., Korhonen, A., and Bentz, C. (2014). A Quantitative Empirical Analysis of the Abstract/Concrete Distinction. *Cognitive Science*, 38:162–177.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Hoppermann, C. and Hinrichs, E. (2014). Modeling Prefix and Particle Verbs in GermaNet. In *Proceedings of the Seventh Global Wordnet Conference*, pages 49–54.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2015). SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 95–105, Beijing, China.

Iacobacci, I., Pilehvar, M. T., and Navigli, R. (2016). Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 897–907, Berlin, Germany.

Jégou, H., Harzallah, H., and Schmid, C. (2007). A contextual Dissimilarity Measure for Accurate and Efficient Image Search. In *Proceedings of the Conference on*

Computer Vision & Pattern Recognition, pages 1–8, Minneapolis, MN, USA.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the International Conference on Multimedia*, pages 675–678, New York, NY, USA.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2016). Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Jones, K. S. (1972). A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28:11–21.

Jurafsky, D. and Martin, J. H. (2017). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Artificial Intelligence. Prentice Hall, 3rd edition. preprint on webpage at `https://web.stanford.edu/~jurafsky/slp3/`.

Jurgens, D. and Klapaftis, I. (2013). SemEval-2013 Task 13: Word Sense Induction for Graded and Non-Graded Senses. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics*, pages 290–299, Atlanta, Georgia, USA.

Kaalep, H.-J. and Muischnek, K. (2002). Using the Text Corpus to Create a Comprehensive List of Phrasal Verbs. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 101–105, Las Palmas de Gran Canaria, Spain.

Kaalep, H.-J. and Muischnek, K. (2006). Multi-word verbs in a Flective Language: the Case of Estonian. In *Proceedings of the Workshop on Multi-word-expressions in a multilingual context*, pages 57–64, Trento, Italy.

Kaalep, H.-J. and Muischnek, K. (2008). Multi-word Verbs of Estonian: a Database and a Corpus. In *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*, pages 23–26.

Kalivoda, Á. (2017). Hungarian Particle Verbs in a Corpus-driven Approach. In *Computational Linguistics and Intelligent Text Processing - 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Proceedings*, page (In press). Springer, Springer.

Kanske, P. and Kotz, S. (2010). Leipzig Affective Norms for German: A reliability Study. *Behavior Research Methods*, 42(4):987–991.

Kempcke, G. (1965). Die Bedeutungsgruppen der Verbalen Kompositionspartikeln an-und auf-in Synchronischer und Diachronischer Sicht. In *Beiträge zur Geschichte* *der deutschen Sprache und Lteratur*, volume 87, pages 392–426.

Khvtisavrishvili, N., Bott, S., and Schulte im Walde, S. (2015). Wie oft schreibt man das zusammen? The Puzzle of Why some Separable Verbs in German are More Separable than Others. In *Proceedings of the 26th International Conference of the German Society for Computational Linguistics and Language Technology*, Duisburg/Essen, Germany.

Kiela, D. and Bottou, L. (2014). Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45.

Kiela, D., Hill, F., Korhonen, A., and Clark, S. (2014). Improving Multi-Modal Representations Using Image Dispersion: Why Less is Sometimes More. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 835–841.

Kiela, D., Verő, A. L., and Clark, S. (2016). Comparing Data Sources and Architectures for Deep Visual Representation Learning in Semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Austin, TX.

Kim, S. N. and Baldwin, T. (2006). Automatic Identification of English Verb Particle Constructions using Linguistic Features. In *Proceedings of the Third ACL-SIGSEM Workshop on Prepositions*, pages 65–72. Association for Computational Linguistics.

Kisselew, M., Padó, S., Palmer, A., and Šnajder, J. (2015). Obtaining a better understanding of distributional models of German derivational morphology. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 58–63, London, UK.

Klein, W. and Geyken, A. (2010). Das Digitale Wörterbuch der Deutschen Sprache (DWDS). Volume 26:79–96.

Kliche, F. (2009). Zur Semantik der Partikelverben auf *ab*. Eine Studie im Rahmen der Diskursepräentationstheorie. Master's thesis, Universität Tübingen.

Kliche, F. (2011). Semantic Variants of German Particle Verbs with *"ab"*. *Leuvense Bijdragen*, 97:3–27.

Koehn, P. and Hoang, H. (2007). Factored Translation Models. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republic.

Kohomban, U. S. and Lee, W. S. (2005). Learning Semantic Classes for Word Sense Disambiguation. In *Proceedings of the 43rd Annual Meeting on Association for Com-*

putational Linguistics, pages 34–41, Ann Arbor, MI.

Köper, M., Kim, E., and Klinger, R. (2017). IMS at EmoInt-2017: Emotion Intensity prediction with Affective norms, Automatically extended Resources and Deep Learning. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–57.

Köper, M., Scheible, C., and Schulte im Walde, S. (2015). Multilingual Reliability and "Semantic" Structure of Continuous Word Spaces. In *Proceedings of the 11th Conference on Computational Semantics*, pages 40–45, London, UK.

Köper, M. and Schulte im Walde, S. (2016). Automatic Semantic Classification of German Preposition Types: Comparing Hard and Soft Clustering Approaches across Features. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, Berlin, Germany.

Köper, M. and Schulte im Walde, S. (2016). Automatically Generated Affective Norms of Abstractness, Arousal, Imageability and Valence for 350 000 German Lemmas. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 2595–2598, Portoroz, Slovenia.

Köper, M. and Schulte im Walde, S. (2016). Distinguishing Literal and Non-Literal Usage of German Particle Verbs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 353–362, San Diego, California, USA.

Köper, M. and Schulte im Walde, S. (2017a). Applying Multi-Sense Embeddings for German Verbs to Determine Semantic Relatedness and to Detect Non-Literal Language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 535–542, Valencia, Spain.

Köper, M. and Schulte im Walde, S. (2017b). Complex Verbs are Different: Exploring the Visual Modality in Multi-Modal Models to Predict Compositionality. In *Proceedings of the 13th Workshop on Multiword Expressions*, pages 200–206, Valencia, Spain.

Köper, M. and Schulte im Walde, S. (2017c). Improving Verb Metaphor Detection by Propagating Abstractness to Words, Phrases and Individual Senses. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 24–30, Valencia, Spain.

Köper, M. and Schulte im Walde, S. (2018). Analogies in Complex Verb Meaning Shifts: The Effect of Affect in Semantic Similarity Models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguis-*

tics: Human Language Technologies, pages 150–156, New Orleans, Louisiana , USA.

Köper, M., Schulte im Walde, S., Kisselew, M., and Padó, S. (2016). Improving Zero-Shot-Learning for German Particle Verbs by using Training-Space Restrictions and Local Scaling. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 91–96, Berlin, Germany.

Kövecses, Z. (2002). *Metaphor: A Practical Introduction*. Oxford University Press, New York.

Krenn, B. and Evert, S. (2001). Can we do better than Frequency? A Case Study on Extracting PP-Verb Collocations. In *Proceedings of the ACL Workshop on Collocations*, Toulouse, France.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

Kühner, N. and Schulte im Walde, S. (2010). Determining the Degree of Compositionality of German Particle Verbs by Clustering Approaches. In *Proceedings of the 10th Conference on Natural Language Processing*, pages 47–56, Saarbrücken, Germany.

Kunze, C. (2000). Extension and Use of GermaNet, a Lexical-Semantic Database. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 999–1002, Athens, Greece.

Köper, M. (2014). Comparing Context-Predicting and Context-Counting Word Representations for Similarity across Words and Relations. Master's thesis, Institut für Maschinelle Sprachverarbeitung, Stuttgart, Germany.

Lahl, O., Göritz, A. S., Pietrowsky, R., and Rosenberg, J. (2009). Using the world-wide web to obtain large-scale word norms: 190,212 ratings on a set of 2,654 german nouns. *Behavior Research Methods*, 41(1):13–9.

Lakoff, G. and Johnson, M. (1980). *Metaphors we live by*. University of Chicago Press.

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

Lapesa, G., Padó, S., Pross, T., and Roßdeutscher, A. (2017). Are Doggies really nicer than Dogs? The Impact of morphological Derivation on Emotional Valence in German. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.

Lau, J. H., Cook, P., McCarthy, D., Newman, D., and Baldwin, T. (2012). Word Sense Induction for Novel Sense Detection. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601, Avignon, France.

Lazaridou, A., Bruni, E., and Baroni, M. G. (2014). Is this a Wampimuk? Cross-Modal Mapping between Distributional Semantics and the Visual World. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1403–1414, Baltimore, Maryland.

Lazaridou, A., Dinu, G., and Baroni, M. (2015a). Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 270–280, Beijing, China.

Lazaridou, A., Marelli, M., Zamparelli, R., and Baroni, M. (2013). Compositional-ly derived Representations of Morphologically Complex Words in Distributional Semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1517–1526, Sofia, Bulgaria.

Lazaridou, A., Pham, N. T., and Baroni, M. (2015b). Combining Language and Vision with a Multimodal Skip-gram Model. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, Denver, Colorado, USA.

Learn German on Lingolia (2018). Trennbare und untrennbare Verben in der deutschen Grammatik.

Lebeth, K. (1992). Zur Analyse von Trennbaren Lokalen Präfixverben in der HPSG. *Bernd Abb und Kai Lebeth (Hgg.), Syntaktische Analysevorschläge zur Behandlung von lokalen Präfixverben in einem System für die Generierung von Wegbeschreibungen, IWBS Report*, (224).

Lechler, A. and Roßdeutscher, A. (2009a). Analysing German Verb-Particle Construction with *auf* in a DRT-based Framework. Technical Report 4, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Lechler, A. and Roßdeutscher, A. (2009b). German Particle Verbs with *auf*. Reconstructing their Composition in a DRT-based Framework. *Linguistische Berichte*, 220:439–478.

Leviant, I. and Reichart, R. (2015). Judgment Language Matters: Multilingual Vector Space Models for Judgment Language Aware Lexical Semantics. *Preprint pubslished on arXiv*, abs/1508.00106.

Levy, O. and Goldberg, Y. (2014a). Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the 18th Conference on Computational Natural Language Learning*, pages 171–180, Maryland, USA, June.

Levy, O. and Goldberg, Y. (2014b). Neural Word Embedding as Implicit Matrix Factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of Computational Linguistics*, 3:211–225.

Li, J. and Jurafsky, D. (2015). Do Multi-Sense Embeddings Improve Natural Language Understanding? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal.

Li, L. and Sporleder, C. (2009). Classifier Combination for Contextual Idiom Detection Without Labelled Data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 315–323, Singapore.

Li, W., Zhang, X., Niu, C., Jiang, Y., and Srihari, R. (2003). An Expert Lexicon Approach to Identifying English Phrasal Verbs. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 513–520. Association for Computational Linguistics.

Lieber, R. and Baayen, H. (1993). *Verbal Prefixes in Dutch: a Study in Lexical Conceptual Structure*, pages 51–78. Springer Netherlands, Dordrecht.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In *Proceedings of the 17th International Conference on Computational Linguistics*, pages 768–774, Montréal, Canada.

Lindner, S. (1983). *A lexico-Semantic Analysis of English Verb Particle Constructions with "out" and "up"*. Indiana University Linguistics Club.

Lipka, L. (1975). Semantic Structure and Word Formation: Verb-Particle Constructions in Contemporary English. *Foundations of Language*, 13(4):593–596.

Litkowski, K. and Hargraves, O. (2007). Semeval-2007 task 06: Word-sense disambiguation of prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 24–29. Association for Computational Linguistics.

Litkowski, K. C. and Hargraves, O. (2005). The preposition project. In *Proceedings of the Second ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, pages 171–179.

Liu, D. (2011). The most frequently used English Phrasal Verbs in American and British English: A Multicorpus Examination. *Tesol Quarterly*, 45(4):661–688.

Lüdeling, A. (1999). *On Particle Verbs and Similar Constructions in German*. Arbeitspapiere des Sonderforschungsbereichs 340 "Sprachtheoretische Grundlagen für die Computerlinguistik". Universität Stuttgart.

Lüdeling, A. (2001). *On German Particle Verbs and Similar Constructions in German*. Dissertations in Linguistics. CSLI Publications, Stanford, CA.

Lüdeling, A. and De Jong, N. (2002). German Particle Verbs and Word-Formation. *Verb-particle explorations*, pages 315–333.

Luong, M.-T., Socher, R., and Manning, C. D. (2013). Better Word Representations with Recursive Neural Networks for Morphology. *Conference on Computational Natural Language Learning*, page 104.

MacQueen, J. (1967). Some Methods of Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Manandhar, S. and Klapaftis, I. P. (2009). SemEval-2010 Task 14: Evaluation Setting for Word Sense Induction & Disambiguation Systems. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 117–122, Stroudsburg, PA, USA. Association for Computational Linguistics.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Markman, A. B. and Gentner, D. (1993). Structural Alignment during Similarity Comparisons. *Cognitive psychology*, 25(4):431–467.

Martin, J. H. (1996). Computational Approaches to Figurative Language. *Metaphor and Symbolic Activity*.

McCallum, A. and Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*.

McCarthy, D., Keller, B., and Carroll, J. (2003). Detecting a Continuum of Compositionality in Phrasal Verbs. In *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 73–80, Sapporo, Japan.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. (2007). Unsupervised Acquisition of Predominant Word Senses. *Computational Linguistics*, 33(4):553–590.

McCarthy, D. and Navigli, R. (2007). SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53, Prague, Czech Republic.

McIntyre, A. (2001). *German Double Particles as Preverbs. Morphology and Conceptual Semantics*. Number 61 in Studien zur deutschen Grammatik. Stauffenburg-Verlag, Tübingen, Germany.

McIntyre, A. (2002). Idiosyncrasy in Particle Verbs. volume 1, page 95. Walter de Gruyter.

Melymuka, M., Lapesa, G., Kisselew, M., and Padó, S. (2017). Modeling Derivational Morphology in Ukrainian. In *IWCS 2017—12th International Conference on Computational Semantics—Short papers*.

Mihalcea, R., Sinha, R., and McCarthy, D. (2010). SemEval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 9–14, Uppsala, Sweden.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013c). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, Lake Tahoe, Nevada, USA.

Mikolov, T., tau Yih, W., and Zweig, G. (2013d). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, GA.

Milajevs, D., Kartsaklis, D., Sadrzadeh, M., and Purver, M. (2014). Evaluating Neural Word Representations in Tensor-Based Compositional Settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 708–719, Doha, Qatar.

Mitchell, T. M. (1997). *Machine Learning*. Computer Science. McGraw-Hill, Boston (MA).

Mohammad, S. M. and Kiritchenko, S. (2015). Using Hashtags to Capture Fine Emotion Categories from Tweets. *Computational Intelligence*, 31(2):301–326.

Mohammad, S. M., Shutova, E., and Turney, P. D. (2016). Metaphor as a Medium for

Emotion: An Empirical Study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*Sem)*, Berlin, Germany.

Morgan, P. S. (1997). Figuring out *figure out*: Metaphor and the Semantics of English Verb-Particle Constructions. *Cognitive Linguistics*, 8(4):327–357.

Muischnek, K., Müürisep, K., and Puolakainen, T. (2013). Estonian Particle Verbs and their Syntactic Analysis. In *Human Language Technologies as a Challenge for Computer Science and Linguistics: 6th Language & Technology Conference Proceedings. December*, pages 7–9.

Müller, S. (2002). Syntax or Morphology: German Particle Verbs Revisited. In Dehé, N., Jackendoff, R., McIntyre, A., and Urban, S., editors, *Verb-Particle Explorations*, Interface Explorations, pages 119–139. Mouton de Gruyter, Berlin, New York.

Müller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, WA, USA.

Murphy, M. (2010). *Lexical Meaning*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Nagy, I. and Vincze, V. (2014). VPCTagger: Detecting Verb-Particle Constructions With Syntax-Based Methods. In *The 10th Workshop on Multiword Expressions*, Gothenburg, Sweden.

Nattinger, J. and DeCarrico, J. (1992). *Lexical Phrases and Language Teaching*. Oxford Applied Linguistics.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2014). Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069, Doha, Qatar.

Neeleman, A. and Schipper, J. (1993). Verbal Prefixation in Dutch: Thematic Evidence for Conversion. In *Yearbook of Morphology 1992*, pages 57–92. Springer.

Neeleman, A. and Weerman, F. (1993). The Balance between Syntax and Morphology: Dutch Particles and Resultatives. *Natural language & linguistic theory*, 11(3):433–475.

Nießen, S. and Ney, H. (2000). Improving SMT Quality with Morpho-Syntactic Analysis. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1081–1085. Association for Computational Linguistics.

Nunberg, G. (1992). Systematic Polysemy in Lexicology and Lexicography. In Tommola, H. and Krista Varantola, T. S.-T. . J. S., editors, *Proceedings of the 5th EU-*

RALEX International Congress, pages 386–396, Tampere, Finland. Tampereen YIiopisto.

Olsen, S. (1986). *Wortbildung im Deutschen: eine Einführung in die Theorie der Wortstruktur*, volume 660. Kröner.

Olsen, S. (1995). Über Präfix-und Partikelverbsysteme. *FAS Papers in linguistics*, 3:86–112.

Olsen, S. (1997). Zur Kategorie Verbpartikel. *Beiträge zur Geschichte der deutschen Sprache und Literatur (PBB)*, 119(1):1–32.

Padó, S. and Lapata, M. (2007). Dependency-Based Construction of Semantic Space Models. *Comput. Linguist.*, 33(2):161–199.

Paivio, A. (1990). *Mental Representations: A Dual Coding Approach*. Oxford Psychology Series. Oxford University Press.

Palatucci, M., Pomerleau, D., Hinton, G., and Mitchell, T. (2009). Zero-Shot Learning with Semantic Output Codes. In *Advances in Neural Information Processing Systems 22*, pages 1410–1418.

Pantel, P. (2003). *Clustering by committee*. PhD thesis.

Paul, H. (1920). Deutsche Grammatik. Bd. IV. *Halle a. Saale: Verlag von Max Niemeyer.(Zitiert auf Seite 18)*.

Pelevina, M., Arefiev, N., Biemann, C., and Panchenko, A. (2016). Making Sense of Word Embeddings. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 174–183, Berlin, Germany.

Pelleg, D. and Moore, A. (1999). Accelerating Exact K-means Algorithms with Geometric Reasoning. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 277–281, New York, NY, USA. ACM.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.

Poitou, J. (2003). Fortbewegungsverben, Verbpartikel, Adverb und Zirkumposition. *Cahiers d'études Germaniques*, 2003:69–84.

Popović, M. (2017). Comparing Language Related Issues for NMT and PBMT between German and English. *The Prague Bulletin of Mathematical Linguistics*, 108(1):209–220.

Prescher, D., Riezler, S., and Rooth, M. (2000). Using a Probabilistic Class-Based Lexicon for Lexical Ambiguity Resolution. In *Proceedings of the 18th International*

Conference on Computational Linguistics, pages 649–655, Saarbrücken, Germany.

Pustejovsky, J. (1995). *The Generative Lexicon*. MIT Press, Cambridge, MA.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1:81–106.

Radovanović, M., Nanopoulos, A., and Ivanović, M. (2010). Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research*, 11:2487–2531.

Rapp, R. (2003). Word Sense Discovery based on Sense Descriptor Dissimilarity. In *Proceedings of the Ninth Machine Translation Summit*, pages 315–322.

Recchia, G. and Louwerse, M. M. (2015). Reproducing Affective Norms with Lexical Co-occurrence Statistics: Predicting Valence, Arousal, and Dominance. *The Quarterly Journal of Experimental Psychology*, 68(8):1584–1598.

Redondo, J., Fraga, I., Padrón, I., and Comesaña, M. (2007). The Spanish Adaptation of ANEW (Affective Norms for English Words). *Behavior Research Methods*, 39(3):600–605.

Reimers, N. and Gurevych, I. (2017). Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 338–348.

Reisinger, J. and Mooney, R. J. (2010). Multi-prototype Vector-space Models of Word Meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 109–117.

Richter, E. (2010). The Acquisition of Prefix and Particle Verbs in German: Evidence from CHILDES. In *LSA Annual Meeting Extended Abstracts*, volume 1, pages 45–1.

Riemsdijk, H. C. v. (1978). *A case study in syntactic markedness : the binding nature of prepositional phrases / H. C. van Riemsdijk*. The Peter de Ridder Press Lisse.

Roller, S. and Schulte im Walde, S. (2013). A Multimodal LDA Model integrating Textual, Cognitive and Visual Modalities. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1146–1157, Seattle, WA.

Rüd, S. (2012). Untersuchung der distributionellen Eigenschaften der Lesarten der Partikel *'auf'* mittels Clustering-Methoden. Diplomarbeit, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A., and Flickinger, D. (2002). Multiword Expressions: A Pain in the Neck for NLP. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico.

Salton, G. (1971). *The SMART Retrieval System&#8212;Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Samuel, A. L. (1959). Some Studies in Machine Learning using the Game of Checkers. *IBM Journal of Research and Development*, pages 71–105.

Schäfer, R. (2015). Processing and Querying Large Web Corpora with the COW14 Architecture. In Bański, P., Biber, H., Breiteneder, E., Kupietz, M., Lüngen, H., and Witt, A., editors, *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*, pages 28 – 34.

Schäfer, R. and Bildhauer, F. (2012). Building Large Corpora from the Web Using a New Efficient Tool Chain. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 486–493, Istanbul, Turkey.

Schiehlen, M. (2003). A Cascaded Finite-State Parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 163–166, Budapest, Hungary.

Schlechtweg, D., Eckmann, S., Santus, E., Schulte im Walde, S., and Hole, D. (2017). German in Flux: Detecting Metaphoric Change via Word Entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 354–367, Vancouver, Canada.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging using Decision Trees. In *Proceedings of the 1st International Conference on New Methods in Language Processing*.

Schmidtke, D., Schröder, T., Jacobs, A., and Conrad, M. (2014). ANGST: Affective norms for German Sentiment Terms, Derived from the Affective Norms for English Words. *Behavior Research Methods*, 46(4):1108–1118.

Schnitzer, D., Flexer, A., and Tomasev, N. (2014). A Case for Hubness Removal in High-Dimensional Multimedia Retrieval. In *Advances in Information Retrieval - 36th European Conference on IR Research*, pages 687–692.

Schottmüller, N. and Nivre, J. (2014). Issues in Translating Verb-Particle Constructions from German to English. In *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*, pages 124–131.

Schulte im Walde, S. (2006). The Syntax-Semantics Interface of German Particle Verbs. Panel discussion at the 3rd ACL-SIGSEM Workshop on Prepositions at the 11th Conference of the European Chapter of the Association for Computational Linguistics.

Schulte im Walde, S., Hätty, A., and Bott, S. (2016). The Role of Modifier and Head Properties in Predicting the Compositionality of English and German Noun-Noun

Compounds: A Vector-Space Perspective. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 148–158, Berlin, Germany.

Schulte im Walde, S., Köper, M., and Springorum, S. (2018). Assessing Meaning Components in German Complex Verbs: A Collection of Source-Target Domains and Directionality. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 22–32, New Orleans, LA, USA.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464.

Sedoc, J., Preotiuc-Pietro, D., and Ungar, L. (2017). Predicting Emotional Word Ratings using Distributional Representations and Signed Clustering. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, EACL, Valencia, Spain.

Shutova, E., Kiela, D., and Maillard, J. (2016). Black Holes and white Rabbits: Metaphor Identification with Visual Features. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 160–170.

Shutova, E., Sun, L., and Korhonen, A. (2010). Metaphor Identification Using Verb and Noun Clustering. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1002–1010.

Shutova, E. and Teufel, S. (2010). Metaphor Corpus Annotated for Source - Target Domain Mappings. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 3225–3261, Valletta, Malta.

Shutova, E., Teufel, S., and Korhonen, A. (2013). Statistical Metaphor Processing. *Computational Linguistics*, 39(2):301–353.

Siegel, S. and Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, Boston, MA.

Silberer, C. and Lapata, M. (2012). Grounded Models of Semantic Representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1423–1433, Jeju Island, Korea. Association for Computational Linguistics.

Silberer, C. and Lapata, M. (2014). Learning Grounded Meaning Representations with Autoencoders. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 721–732, Baltimore, Maryland.

Smith, S. L., Turban, D. H. P., Hamblin, S., and Hammerla, N. Y. (2017). Offline Bilingual Word Vectors, Orthogonal Transformations and the Inverted Softmax.

In *Proceedings of 5th International Conference on Learning Representations*, Toulon, France.

Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*, pages 4444–4451.

Sporleder, C. and Li, L. (2009). Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 754–762, Athens, Greece.

Springorum, S. (2009). Zur Semantik der Partikelverben mit *an*. Eine Studie zur Konstruktion ihrer Bedeutung im Rahmen der Diskursrepräsentationstheorie. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Springorum, S. (2011). DRT-based Analysis of the German Verb Particle *"an"*. *Leuvense Bijdragen*, 97:80–105.

Springorum, S., Schulte im Walde, S., and Roßdeutscher, A. (2012). Automatic Classification of German an Particle Verbs. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 73–80, Istanbul, Turkey.

Springorum, S., Schulte im Walde, S., and Roßdeutscher, A. (2013a). Sentence Generation and Compositionality of Systematic Neologisms of German Particle Verbs. Talk at the 5th Conference on Quantitative Investigations in Theoretical Linguistics.

Springorum, S., Utt, J., and Schulte im Walde, S. (2013b). Regular Meaning Shifts in German Particle Verbs: A Case Study. In *Proceedings of the 10th International Conference on Computational Semantics*, pages 228–239, Potsdam, Germany.

Steen, G. J., Dorst, A. G., Herrmann, J. B., Kaal, A. A., and Krennmayr, T. (2010). *Cognitive Linguistics*, 21(4):765–796.

Steiger, J. H. (1980). Tests for Comparing Elements of a Correlation Matrix. *Psychological Bulletin*.

Stevenson, R., Mikels, J., and James, T. (2007). Characterization of the Affective Norms for English Words by Discrete Emotional Categories. *Behavior Research Methods*, 39(4):1020–1024.

Stevenson, S., Fazly, A., and North, R. (2004). Statistical Measures of the Semi-Productivity of Light Verb Constructions. In *Proceedings of the 2nd Workshop on Multiword Expressions: Integrating Processing*, pages 1–8, Barcelona, Spain.

Stiebels, B. (1996). *Lexikalische Argumente und Adjunkte. Zum semantischen Beitrag von verbalen Präfixen und Partikeln*. Akademie Verlag, Berlin.

Stiebels, B. and Wunderlich, D. (1994). Morphology feeds Syntax: The Case of Particle Verbs. *Linguistics*, 32:913–968.

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 8–15, Sapporo, Japan.

Svenonius, P. (1996). The Verb-Particle Alternation in the Scandinavian Languages. *University of Tromsø*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition*.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2004). Hierarchical Dirichlet Processes. *Journal of the American Statistical Association, Volume 101*.

Tomasev, N. (2014). *The Role Of Hubness in High-dimensional Data Analysis*. PhD thesis.

Tsvetkov, Y., Boytsov, L., Gershman, A., Nyberg, E., and Dyer, C. (2014). Metaphor Detection with Cross-Lingual Model Transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 248–258, Baltimore, Maryland.

Tsvetkov, Y., Mukomel, E., and Gershman, A. (2013). Cross-lingual Metaphor Detection Using Common Semantic Features. In *Proceedings of the Workshop on Metaphor in NLP*, pages 45–51, Atlanta, USA.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word Representations: A Simple and General Method for Semisupervised Learning. In *In ACL*, pages 384–394, Uppsala, Sweden.

Turney, P. D. (2006). Similarity of Semantic Relations. *Computational Linguistics*, 32(3):379–416.

Turney, P. D. (2012). Domain and Functions: A Dual-Space Model of Semantic Relations and Compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Turney, P. D. and Littman, M. L. (2003). Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems*, pages 315–346.

Turney, P. D., Neuman, Y., Assaf, D., and Cohen, Y. (2011). Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, UK.

Turney, P. D. and Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Tversky, A. (1977). *Psychological review*, 84(4):327.

Twain, M. (1880). *The awful German language*. BVK.

Vankrunkelsven, H., Verheyen, S., Deyne, S. D., and Storms, G. (2015). Predicting Lexical Norms Using a Word Association Corpus. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, Pasadena, California, USA.

Venturi, G., Montemagni, S., Marchi, S., Sasaki, Y., Thompson, P., McNaught, J., and Ananiadou, S. (2009). Bootstrapping a Verb Lexicon for Biomedical Information Extraction. In Gelbukh, A., editor, *Linguistics and Intelligent Text Processing*, pages 137–148. Springer, Heidelberg.

Vigliocco, G., Kousta, S.-T., Della Rosa, P. A., Vinson, D. P., Tettamanti, M., Devlin, J. T., and Cappa, S. F. (2014). The Neural Representation of Abstract Words: the Role of Emotion. *Cerebral Cortex*, pages 1767–1777.

Villavicencio, A. (2005). The Availability of Verb-Particle Constructions in Lexical Resources: How much is enough? *Computer Speech and Language*, 19:415–432.

Volk, M., Clematide, S., Graën, J., and Ströbel, P. (2016). Bi-Particle Adverbs, PoS-Tagging and the Recognition of German Separable Prefix Verbs. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016), Bochum*, pages 297–305.

Võ, M., Conrad, M., Kuchinke, L., Urton, K., Hofmann, M., and Jacobs, A. (2009). The Berlin Affective Word List Reloaded (BAWL-R). *Behavior Research Methods*, 41(2):534–538.

Võ, M., Jacobs, A., and Conrad, M. (2006). Cross-validating the Berlin Affective Word List. *Behavior Research Methods*, 38(4):606–609.

Wang, S., Durrett, G., and Erk, K. (2018). Modeling Semantic Plausibility by Injecting World Knowledge. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 303–308, New Orleans, Louisiana , USA.

Warriner, A. B., Kuperman, V., and Brysbaert, M. (2013). Norms of Valence, Arousal, and Dominance for 13,915 English Lemmas. *Behavior Research Methods*, 45(4):1191–1207.

Wasow, T., Perfors, A., and Beaver, D. (2005). The Puzzle of Ambiguity. *Morphology and the web of grammar: Essays in memory of Steven G. Lapointe*, pages 265–282.

Weelden, L., Maes, A., Schilperoord, J., and Swerts, M. (2012). How object shape affects visual metaphor processing. 59:1–8.

Weller, M., Schulte im Walde, S., and Fraser, A. (2014). Using Noun Class Information to model Selectional Preferences for Translating Prepositions in SMT. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pages 275–287, Vancouver, Canada.

Wendlandt, L., Kummerfeld, J. K., and Mihalcea, R. (2018). Factors Influencing the Surprising Instability of Word Embeddings. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2092–2102, New Orleans, Louisiana , USA.

Wiemer-Hastings, P. and Zipitria, I. (2001). Rules for syntax, vectors for semantics. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Erlbaum.

Wilks, Y. (1975). A Preferential, Pattern-seeking, Semantics for Natural Language Inference. *Artificial Intelligence*, 6(1):53–74.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition.

Wittmann, M., Köper, M., and Schulte im Walde, S. (2017). Exploring Soft-Clustering for German (Particle) Verbs across Frequency Ranges. In *Proceedings of the 12th International Conference on Computational Semantics*, Montpellier, France.

Wittmann, M., Weller, M., and Schulte im Walde, S. (2014). Automatic Extraction of Synonyms for German Particle Verbs from Parallel Data with Distributional Similarity as a Re-Ranking Feature. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 1430–1437, Reykjavik, Iceland.

Wittmann, M., Weller-Di Marco, M., and Schulte im Walde, S. (2016). Graph-based Clustering of Synonym Senses for German Particle Verbs. In *Proceedings of the 12th Workshop on Multiword Expressions*, pages 38–43, Berlin, Germany.

Yu, M. and Dredze, M. (2015). Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Zeller, B., Šnajder, J., and Padó, S. (2013). DErivBase: Inducing and Evaluating a Derivational Morphology Resource for German. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1201–1211, Sofia, Bulgaria.

Zeller, J. (1997). Against overt Particle Incorporation. *University of Pennsylvania Working Papers in Linguistics*, 4(2):19.

Zeller, J. (2001a). How Syntax restricts the Lexicon: Particles as Thematic Predicates. *Linguistische Berichte*, 188:461–94.

Zeller, J. (2001b). *Particle Verbs and Local Domains*. Linguistik Artuell/Linguistics

Today Series. J. Benjamins.

Zesch, T. and Gurevych, I. (2006). Automatically creating Datasets for Measures of Semantic Relatedness. In *COLING/ACL 2006 Workshop on Linguistic Distances*, pages 16–24, Sydney, Australia.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: an Efficient Data Clustering Method for very Large Databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 103–114.

Zhong, C.-B. and Leonardelli, G. J. (2008). Cold and Lonely: Does Social Exclusion Literally Feel Cold? *Psychological Science*, 19(9):838–842.

Zhong, C.-B. and Liljenquist, K. (2006). Washing away your Sins: Threatened Morality and Physical Cleansing. *Science*, 313(5792):1451–1452.

Zipf, G. K. (1949). *Human Behaviour and the Principle of Least-Effort*. Addison-Wesley, Cambridge.

8 Supplementary Material

Class name	Particle	Size
Ab impliziert d. Entfernung d. Themas	ab	74
abdrängen, abfeuern, abführen, abgraben, abhängen, abholen, abhusten, abkommandieren, ablassen, ableiten, ablenken, abpumpen, abräumen, absaugen, abschicken, abschieben, abschießen, abschmettern, abschütteln, abschütten, abschwemmen, absenden, abstoßen, abtragen, abtransportieren, abbeißen, abbekommen, abbrechen, abessen, abfeilen, abfressen, abgeben, abhacken, abhalftern, abhängen, abhauen, abhäuten, abheben, abknabbern, abkoppeln, abkriegen, abladen, ablegen, ablösen, abmachen, abmontieren, abnabeln, abnagen, abpflücken, abrasieren, abreißen, abrupfen, absägen, absahnen, abscheren, abschlagen, abschminken, abschneiden, abschöpfen, abschrauben, abschroten, abspalten, abspanen, absträngen, abstreifen, abtakeln, abtrennen, abtreten, abtrinken, abzapfen, abzäumen, abziehen, abzupfen, abzwacken		
Lokalisierung eines Themas unterhalb einer Ebene	ab	25
abducken, abfahren, abfallen, abfieren, abflachen, abgleiten, abglitschen, abgrätschen, abhängen, abkippen, abladen, abrunden, abrutschen, absacken, absaufen, abseilen, absenken, absetzen, absinken, abspringen, absteigen, abstürzen, abtauchen, abteufen, abwerfen		
Ab impliziert eine Abnahme	ab	23
abblassen, abblenden, abbremsen, abdämpfen, abdrosseln, abdunkeln, abebben, abflauen, abhungern, abklingen, abkühlen, abmildern, abnehmen, abqualifizieren, abrüsten, abschätzig, abschlaffen, abschmelzen, abschwächen, abschwingen, abwerten, abwiegeln, abschwellen		
Oberfläche oder eine Säuberung	ab	21
abbauen, abbrausen, abbeizen, abblättern, abbürsten, abduschen, abfegen, abfeilen, abfrottieren, abgrasen, abhobeln, abholzen, abkratzen, abnagen, abschaben, abschrubben, abspülen, abstauben, abstreichen, abwaschen, abwischen		
Ab impliziert Besitzwechsel	ab	20
abfordern, abgewinnen, abjagen, abkaufen, abknöpfen, ablisten, ablocken, ablösen, abluchsen, abnehmen, abnötigen, abpressen, abringen, abschwatzen, abspielen, abtausch, abtrotzen, abverlangen, abwechseln, abzocken		
Ab impliziert Handlung des sich Entfernens	ab	19
abbrausen, abdampfen, abdriften, abdüsen, abfahren, abfliegen, abgehen, abhauen, abläuten, ablegen, abmarschieren, abrauschen, abreisen, abrücken, abspringen, abschwirren, abtanzen, abwandern, abzischen		
Kanals bzw. Durchgang verhindern	ab	18
abbinden, abdrehen, abgehackt, abklemmen, abschnüren, abblocken, abfangen, abhalten, ablehnen, ableugnen, abschrecken, abstreiten, abwehren, abweisen, abwimmeln, abriegeln, abschließen, absperren		
Ab bedeutet Kopie	ab	17
abbilden, abfärben, abfotografieren, abgucken, abkonterfeien, abkupfern, ablesen, ablichten, abmalen, abpausen, abschauen, abschreiben, abspiegeln, abstammen, abtippen, abtönen, abzeichnen		
Abnahme mereologischer Teile	ab	17
abarbeiten, abbezahlen, abbummeln, abfischen, abgrasen, abkämmen, abklappern, ableisten, abreiten, absingen, absitzen, abspielen, abspulen, abstottern, absuchen, abweiden, abzahlen		

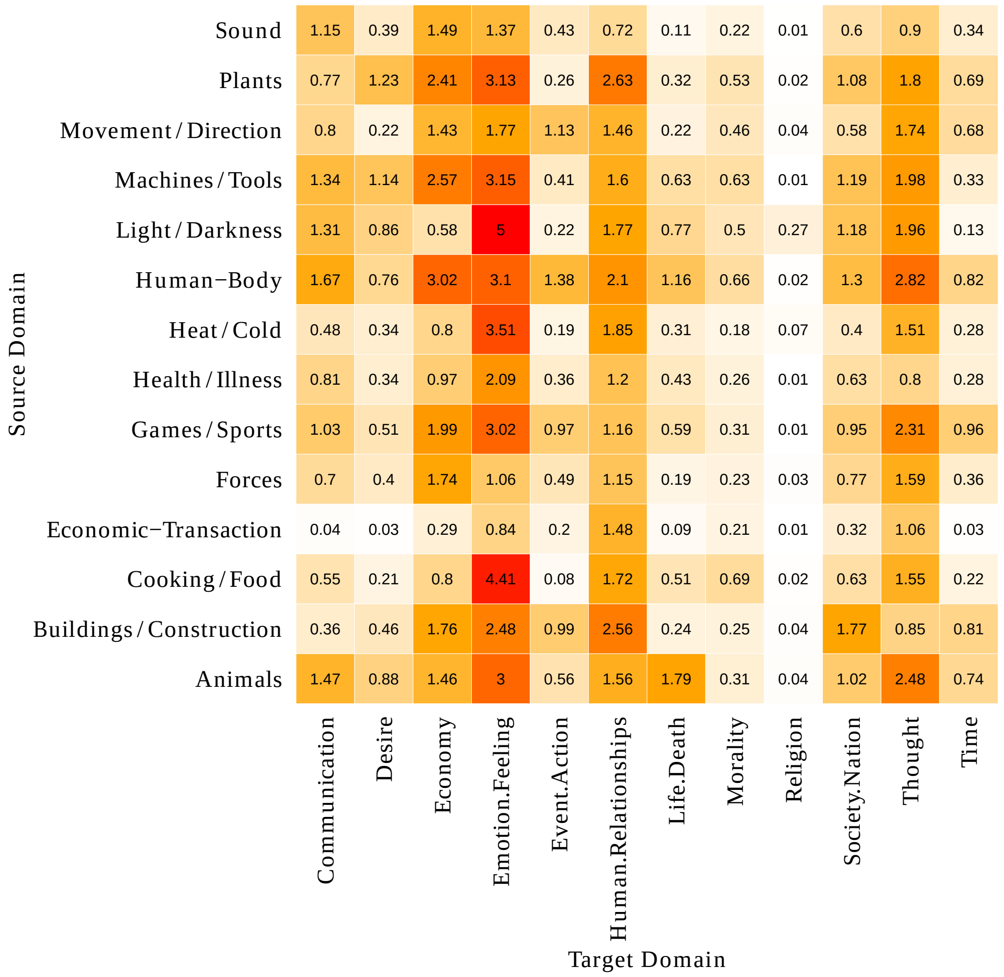
Table 8.1: Reduced classes from the manually created verb classes for 'ab', subset from Kliche (2009)

Class name	**Particle**	**Size**
An impliziert topologischen Kontakt	an	71
anlegen, anmalen, anpinseln, anstreichen, ansprühen, ansetzen, anstecken, anbauen, anbinden, anhaken, anklammern, anketten, annageln, annieten, anlöten, anzwecken, anmontieren, anhängen, ankuppeln, ankleben, anleimen, anflicken, anpassen, anziehen, anstehen, anschließen, anklopfen, ansiedeln, anfassen, anstoßen, angreifen, anlehnen, anhaften, anrempeln, anpflanzen, anhacken, anknüpfen, anstellen, anfallen, ankreuzen, anecken, anbandeln, anfahren, anwachsen, anschlagen, anfuttern, ankleiden, anbohren, anbringen, andrängen, anfühlen, angehören, anheften, anknüpfen, ankoppeln, ankuppeln, anliegen, anprobieren, anbandeln, anrechnen, anreihen, ansaugen, anschmieden, anschmieren, anschnallen, anseilen, ansetzen, anspannen, anspringen, antreffen, anfressen		
Richtung Kommunikationsversuch	an	36
ansprechen, anlachen, angrinsen, anschnauzen, anbrüllen, anranzen, anknurren, anzischen, anfauchen, anreden, anquatschen, anrufen, anflehen, anschweigen, anschreien, anhupen, anklingeln, anfunkeln, anhupen, anpflaumen, anmachen, anspielen, anschnauzen, anmelden, anschwärzen, anleiten, anprangern, anhören, anspielen, andonnern, anfeinden, anpöbeln, anpumpen, anschreiben, anmeckern, anblinzeln		
An markiert Initiierung eines Ereignisses	an	35
ankurbeln, anlaufen, antreiben, anblasen, anfachen, anwerfen, antreten, anmachen, anbrennen, anheizen, anstimmen, anspornen, anreizen, anzapfen, anziehen, anpfeifen, anläuten, anstiften, anregen, anspielen, anfangen, anrichten, anbahnen, anfreunden, antreten, anheuern, angehen, anheben, anhalten, anschicken, anspannen, anspringen, anstoßen, anzetteln, antun		
An impliziert Partitiv-Interpretation	an	21
anrösten, anfeilen, anfressen, anbrechen, anblättern, anreißen, anstechen, ansengen, anbraten, anknabbern, anschneiden, anbräunen, ankippen, anheben, ansteigen, anlupfen, anbeißen, anbohren, anschwärzen, anzahlen, anzapfen,		
impliziert Richtung	an	20
anblicken, anhauchen, anhimmeln, anpeilen, anstarren, anlügen, anstreben, anpeilen, ansehen, angucken, anvisieren, anfahren, anpaddeln, anschleichen, anschwimmen, anfliegen, annähern, anrennen, ansegeln, anziehen		
Richtung mit Deixis	an	20
anmarschieren, ankommen, anrücken, anreiten, anbrausen, anflitzen, anrollen, anrumpeln, antraben, anlangen, anhumpeln, ankriechen, anpirschen, anreisen, antanzen, anfliegen, anlangen, anrennen, anschwärmen, anstürmen		
An impliziert Besitzwechsel	an	13
anschwatzen, andrehen, anliefern, anmieten, anwerben, aneignen, anschaffen, anärgern, anreichen, ankaufen, anvertrauen, anlernen, annehmen		
An impliziert Intensivierung	an	11
anziehen, anheizen, anstrengen, anhetzen, anpeitschen, anfeuern, anspornen, anstacheln, ansteigen, antreiben, anregen		

Table 8.2: Reduced classes from the manually created verb classes for ‘an’, subset from Springorum (2011)

<table>
<tr><th>Class name</th><th>Particle</th><th>Size</th></tr>
<tr><td>Auf bedeutet offen</td><td>auf</td><td>72</td></tr>
<tr><td colspan="3">aufsein, aufbleiben, aufstehen, aufklaffen, aufhalten, auflassen, auffliegen, aufbrechen, aufgehen, aufplatzen, aufreißen, aufschnappen, aufschnellen, aufschwingen, aufspringen, aufblühen, aufkrachen, aufmachen, aufbeißen, aufbekommen, aufbiegen, aufblättern, aufbohren, aufbrechen, aufdrehen, aufdrücken, auffeilen, auffetzen, aufhacken, aufkauen, aufklappen, aufklinken, aufklopfen, aufknacken, aufkratzen, aufreiben, aufreißen, aufritzen, aufscharren, aufscheuern, aufschieben, aufschießen, aufschlagen, aufschlitzen, aufschnallen, aufschneiden, aufschrauben, aufschürfen, aufschwingen, aufspalten, aufspannen, aufsprengen, aufstemmen, aufstoßen, auftrennen, aufziehen, aufwehen, aufdecken, aufschliessen, aufriegeln, aufsperren, aufknüpfen, auffalten, aufpacken, aufflechten, aufknoten, aufknöpfen, aufkorken, aufschnüren, aufbinden, aufrollen, aufwickeln</td></tr>
<tr><td>Verbesserung und Vergrößerung</td><td>auf</td><td>48</td></tr>
<tr><td colspan="3">aufdrehen, aufheizen, aufwerten, aufbrisen, aufstocken, aufholzen, aufforsten, aufsiedeln, aufbetten, aufrunden, aufladen, aufstufen, aufbacken, aufpolieren, aufmischen, aufbauen, aufbeizen, aufbinden, aufbraten, aufbürsten, aufkämmen, aufbügeln, auffärben, aufforsten, aufholzen, auffüllen, auftanken, aufpolstern, aufpudern, aufarbeiten, aufschwärzen, aufrühren, auftunen, aufhellen, aufwärmen, aufbessern, auflockern, aufbereiten, aufhöhen, aufstylen, auftakeln, aufbrezeln, aufdonnern, aufmachen, aufmotzen, aufmöbeln, aufputzen, aufpeppen</td></tr>
<tr><td>Auf bedeutet Aufwärts bewegung</td><td>auf</td><td>31</td></tr>
<tr><td colspan="3">aufbranden, aufbrausen, aufbrodeln, aufdampfen, auffahren, aufflattern, auffliegen, auffrieren, aufgehen, auflaufen, aufschnellen, aufschrecken, aufschwimmen, aufschwingen, aufspritzen, aufsprudeln, aufsprühen, aufsteigen, aufstieben, aufstreben, auftauchen, aufwirbeln, aufwogen, aufziehen, aufzüngeln, aufrauschen, aufbrausen, aufschießen, aufwallen, aufkochen, aufwölben</td></tr>
<tr><td>Auf impliziert Wahrnehmung</td><td>auf</td><td>25</td></tr>
<tr><td colspan="3">aufspüren, aufstöbern, auftreiben, aufsuchen, auffinden, auflegen, aufsagen, aufzeigen, aufdeuten, aufweisen, aufkommen, aufkreuzen, aufziehen, auffallen, auffahren, aufgehen, aufbrauen, aufliegen, auffallen, auftanzen, auftreten, aufblitzen, aufleuchten, aufglimmen, aufschreien</td></tr>
<tr><td>Auf für Wachstum</td><td>auf</td><td>21</td></tr>
<tr><td colspan="3">aufsprießen, aufwachsen, aufschießen, aufschütten, auftürmen, aufbringen, aufstapeln, aufhäufen, aufhocken, aufschottern, aufsanden, aufteeren, aufschwellen, aufquellen, aufbauschen, aufblasen, aufschäumen, aufblühen, aufplustern, aufpusten, aufschwemmen</td></tr>
<tr><td>Support (ground, subject)</td><td>auf</td><td>19</td></tr>
<tr><td colspan="3">auftreten, aufprallen, auftreffen, aufkommen, aufkrachen, aufklatschen, auffahren, auflaufen, aufschlagen, aufstampfen, auftippen, aufspringen, aufsetzen, aufstellen, aufbügeln, aufdrücken, auflegen, aufmalen, auftragen</td></tr>
<tr><td>Etwas aufnehmen oder hochnehmen</td><td>auf</td><td>16</td></tr>
<tr><td colspan="3">aufnehmen, aufgreifen, auffischen, aufsammeln, aufheben, auflesen, aufpicken, aufsaugen, auflecken, auftupfen, aufschlecken, aufdippen, auffegen, aufkehren, aufwischen, aufsuchen</td></tr>
<tr><td>Auf bedeutet alle</td><td>auf</td><td>16</td></tr>
<tr><td colspan="3">auffressen, auffuttern, auffüttern, aufkaufen, aufknabbern, auflutschen, aufrauchen, aufschlucken, auftragen, aufzehren, auftrinken, aufarbeiten, aufopfern, aufgehen, aufräumen, auflösen</td></tr>
<tr><td>Vertikale Position</td><td>auf</td><td>11</td></tr>
<tr><td colspan="3">aufkommen, aufspringen, aufstehen, aufstellen, aufrichten, aufsetzen, aufrecken, aufbäumen, aufraffen, aufrappeln, aufhelfen</td></tr>
<tr><td>Widerstand gegen etwas Höheres</td><td>auf</td><td>11</td></tr>
<tr><td colspan="3">aufbegehren, auffahren, aufstehen, aufmucken, auflehnen, aufbringen, aufreizen, aufhetzen, aufmischen, aufstacheln, aufputschen</td></tr>
<tr><td>Auf bedeutet Partition</td><td>auf</td><td>9</td></tr>
<tr><td colspan="3">aufteilen, aufschlüsseln, aufgliedern, auffächern, aufbröckeln, aufsplittern, aufsplitten, aufspalten, aufschneiden</td></tr>
</table>

Table 8.3: Reduced classes from the manually created verb classes for 'auf', subset from Lechler and Roßdeutscher (2009a)



	Communication	Desire	Economy	Emotion.Feeling	Event.Action	Human.Relationships	Life.Death	Morality	Religion	Society.Nation	Thought	Time
Sound	1.15	0.39	1.49	1.37	0.43	0.72	0.11	0.22	0.01	0.6	0.9	0.34
Plants	0.77	1.23	2.41	3.13	0.26	2.63	0.32	0.53	0.02	1.08	1.8	0.69
Movement/Direction	0.8	0.22	1.43	1.77	1.13	1.46	0.22	0.46	0.04	0.58	1.74	0.68
Machines/Tools	1.34	1.14	2.57	3.15	0.41	1.6	0.63	0.63	0.01	1.19	1.98	0.33
Light/Darkness	1.31	0.86	0.58	5	0.22	1.77	0.77	0.5	0.27	1.18	1.96	0.13
Human−Body	1.67	0.76	3.02	3.1	1.38	2.1	1.16	0.66	0.02	1.3	2.82	0.82
Heat/Cold	0.48	0.34	0.8	3.51	0.19	1.85	0.31	0.18	0.07	0.4	1.51	0.28
Health/Illness	0.81	0.34	0.97	2.09	0.36	1.2	0.43	0.26	0.01	0.63	0.8	0.28
Games/Sports	1.03	0.51	1.99	3.02	0.97	1.16	0.59	0.31	0.01	0.95	2.31	0.96
Forces	0.7	0.4	1.74	1.06	0.49	1.15	0.19	0.23	0.03	0.77	1.59	0.36
Economic−Transaction	0.04	0.03	0.29	0.84	0.2	1.48	0.09	0.21	0.01	0.32	1.06	0.03
Cooking/Food	0.55	0.21	0.8	4.41	0.08	1.72	0.51	0.69	0.02	0.63	1.55	0.22
Buildings/Construction	0.36	0.46	1.76	2.48	0.99	2.56	0.24	0.25	0.04	1.77	0.85	0.81
Animals	1.47	0.88	1.46	3	0.56	1.56	1.79	0.31	0.04	1.02	2.48	0.74

Figure 8.1: (Literal) Source → (Non-Literal) Target Domain shifts across all Particles. Un-weighted (not ppmi)