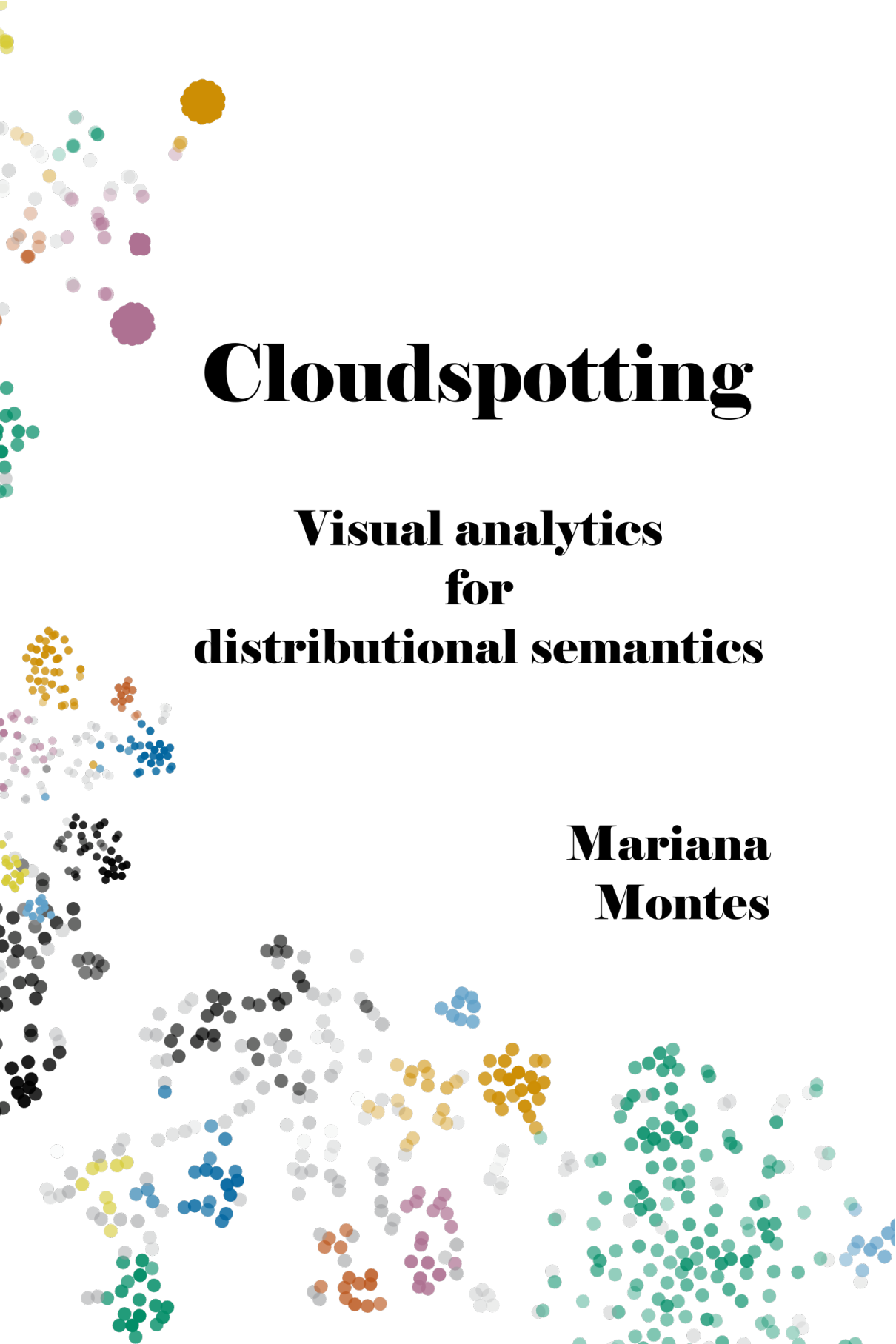

Cloudspotting

Visual analytics for distributional semantics

Mariana Montes

UNIVERSITY OF LEUVEN
FACULTY OF ARTS
DEPARTMENT OF LINGUISTICS

Cloudspotting

Visual analytics for distributional semantics

Thesis presented in partial fulfillment of the requirements
for the degree of Doctor in Linguistics

Mariana Montes

Supervisor: Prof. Dr. Dirk Geeraerts
Co-supervisor: Prof. Dr. Dirk Speelman
Co-supervisor: Prof. Dr. Benedikt Szmrecsanyi

Leuven, July 2021

To Miguel and Patricia

Contents

List of Tables

List of Figures

Preface

The research described in this dissertation is part of the Nephological Semantics[1] research project at the QLVL research group in KU Leuven, which aims to develop tools for large-scale corpus-based semantic analysis. A core aspect of the project involves representing semantic structure with distributional models, a computational tool that currently requires a deeper understanding of its inner workings and how its results relate to cognitive theories of meaning.

Context-counting distributional models represent words[2] as vectors of co-occurrence frequencies in a multidimensional space (Turney & Pantel 2010, Lenci 2018). Basically, a word is represented by its association strength to other words. They can be generated at both type and token level (Heylen, Speelman & Geeraerts 2012, Heylen et al. 2015, De Pascale 2019). At type level, two words are represented as more similar if they are attracted to the same contextual features (e.g. other words) and repelled by the same contextual features. This should allow us to identify semantic fields and other relationships between words, but collapses the full range of contexts of each word into one representation. At token level, instead, we look at individual occurrences and define them as more similar if the words in their contexts are attracted to and repelled by the same contextual features. This way we should be able to map the internal variation of the behaviour of individual words, i.e. their semasiological structure.

Within the larger Nephological Semantics project, this particular work package is dedicated to the understanding of token-level distributional models as a tool for the study of polysemy. Concretely, I explored a number of parameter settings for the models (i.e. ways of defining the context used to represent each token) and their impact on the resulting representation, by means of visual analytics. Manually annotated sense tags were used as a heuristic, but without considering them a golden standard. Instead, the aim was to map parameter settings to various semantic phenomena coded in the annotations, such as meaning granularity (e.g. distinguishing homonyms and senses within the homonyms). The distributional models, which take the form of large matrices, can be reduced to two dimensions via different methods, such as t-SNE (van der Maaten & Hinton 2008, van der Maaten 2014). These coordinates can then be mapped onto a scatterplot, resulting in a variety of shapes, which we call *clouds*.

[1] https://www.arts.kuleuven.be/ling/qlvl/projects/current/nephological-semantics

[2] The term *word* is used very loosely here to encompass different possible definitions.

The workflow was applied to a set of 32 Dutch nouns, verbs and adjectives exhibiting a range of semantic phenomena. For each of them, 240-320 concordance lines were extracted from a corpus of Dutch and Flemish newspapers, annotated and modelled. The combination of parameter settings, some of which included syntactic information, resulted in 200-212 different models per lemma. The models were clustered with Partition Around Medoids (Kaufman & Rousseeuw 1990, Maechler et al. 2021) so that a manageable, representative set could be explored in more depth, in particular visualizing their t-SNE representations.

The contributions of this dissertation are twofold. On the one hand, the exploration of the possibilities and limits of distributional models to lexicological research resulted in warnings, suggestions and guidelines for practical studies. In other words, it offers an assessment and interpretation of distributional models from the perspective of descriptive linguistics. On the other hand, it presents a visualization tool designed for the exploration of token-level distributional models from such a perspective (Montes & QLVL 2021). Its interactive quality makes it challenging to describe it adequately in a printed text, so I would strongly recommend visiting it in its virtual home[3] and explore it.

[3]https://qlvl.github.io/NephoVis/

Acknowledgements

The words in these pages, the thoughts they try to convey, are the result of years of thinking, discussing, learning. My voice weaves them together, but it draws from many sources that have encouraged my growth, stood by me, and fed my curiosity, passion and enthusiasm for everything that makes up this text.

For their support and their ideas, I would like to thank my supervisors Dirk Geeraerts, Dirk Speelman and Benedikt Szmrecsanyi. Thank you for trusting me, for allowing me to be part of this amazing research project. Dirk Geeraerts deserves a special acknowledgement for all his patience and trust. It has been an honour to share such interesting, long discussions on semantics and research. Every time we talked I became more excited and passionate about my research, more confident and happy. Hartelijk bedankt. I am also grateful to Freek Van de Velde, Tim Van de Cruys, Martin Hilpert and Thomas Herbst for their willingness to participate in the jury.

Research is eminently collaborative, but it can feel lonely. I will always appreciate the company and support of my colleagues at the Linguistics Department at KU Leuven, especially my research group, QLVL. In particular, I would like to thank the members of the Nephological Semantics project, with whom I shared so much of the excitement and frustrations of our common project. I'd like to acknowledge Tao Chen for his amazing work in the Python code and helping me understand it and Stefania Marzo for always making me feel like I had done something right. For the past year and a half, the weekly meetings with Dirk and Dirk, Kris, Karlien, Stefano and Weiwei have been intense but have also kept me grounded. I was constantly learning, rethinking, trying new things. I am grateful for your collaboration and your advice, for being there and letting me know that this research matters. Kris has been an unstoppable source of ideas, Karlien such a great partner for spontaneous experiments, and Stefano so helpful with our shared methodological obstacles (and thank you for proofreading!).

Some colleagues become friends. The first is Danqing, who shares with me the experience of studying and making a life far from home, and who has kindly agreed to do some proofreading in spite of her busy schedule. I am also immensely grateful to Marlieke and Pedro, not only for their proofreading efforts and their invaluable help with the translations of the examples but also, and most importantly, for their company and their friendship. Thank you Ola, Caro, Araceli and Manuel for standing by me even from a distance. To Paula,

thank you for challenging me, encouraging me, making me a better researcher and a better person.

From friends we move to family. I honestly wouldn't be where I am today if it weren't for my parents. They have taught me to learn and to teach, to face challenges, to not fear mathematics or programming. My siblings are my strength, thank you for being there, thank you for the mates, the talks, the hugs. My cousin Alex deserves a special acknowledgement for holding my hand while I discovered web design and helping me disentangle a whole new field. Muchas gracias, che.

Finally, I would like to thank my partner, Taihou, who has kept me alive while I was finishing this text, and has always listened to my ramblings, held me when I fell, rejoiced with me when I succeeded.

To all of you, I am forever grateful. This is thanks to you, I hope to make you proud.

Chapter 1

Introduction

If meaning is found and created in use, and corpora are language in use, can we find meaning in corpora? The field of usage-based semantics is large and rich, so the answer to this question is clearly positive. Corpora offer an immense amount of usage data on which to carry analyses, even if they barely scratch the surface of the amount of language that is actually produced — it is desirable and tempting to tap into this vast ocean to obtain the most detailed, the most reliable, the most thorough information. But there is a crucial bottleneck when it comes to semantic analysis: annotation is time- and energy-consuming. As long as we cannot instruct an automatic system to disambiguate each word in a corpus — like we do to tokenize and lemmatize, i.e. to identify what counts as a word and what its root is, or even to assign parts of speech or syntactic relations — semantic annotation is performed by humans. Humans are slower than computers; we get tired, we get confused, we need to eat and think of things beyond semantic annotation as weel. We also disagree sometimes — what is a sense? Are these two things *really* the same?

Automatic disambiguation systems do exist. Word Sense Disambiguation is an important task within Natural Language Processing (NLP). The notion of *task* is of crucial importance here: NLP algorithms are typically concerned with concrete applications and are evaluated in terms of those applications. There exists a correct answer that the algorithm must return. This is not so directly applicable to the situation of lexicological and lexicographical research — the study of the meanings of words and their relationships — especially from a Cognitive Linguistics point of view, where hard, dichotomous answers are rare. But let's suppose for a moment that we can conciliate both approaches, and what counts as *the* answer from an NLP point of view is *an* answer from the lexicological perspective. Then we could use automatic disambiguation procedures to make the heavy lifting of semantic annotation of our growing body of corpus data and use their results for a partial description of language. As long as we know *which* answer the NLP algorithm is returning or, better yet, how to ask what we want to know. Maybe tuning the algorithm for outputs that from an NLP point of view would be *wrong* can result in complementary answers for a richer lexicological description. Such a qualitative perspective,

trying to interpret not just *whether* the computational model matches a target but also *how* or *why* it does (not), also requires appropriate analytical tools. One such tool represents the internal semantic structure of an item, derived from computational models, as a 2D scatterplot where instances occurring in similar context are shown together, forming clusters or *clouds*.

This dissertation is concerned with the application of distributional methods to lexicological research and their exploration by means of visual analytics. The methodology will be tested and illustrated with a set of 32 Dutch lemmas, of which concordance lines will be extracted from a corpus of newspapers. Distributional models, developed within the field of Computational Linguistics, will be introduced in Section 1.1. In Section 1.2 we will discuss their relevance in Cognitive Semantics and Section 1.3 will offer an overview of the visual analytics dimension. The study described here is part of a larger research project within the Quantitative Lexicology and Variational Linguistics research group (QLVL) at KU Leuven. A brief history of the project and how this dissertation fits in it will be offered in Section 1.4. Finally, Section 1.5 will present the structure of the dissertation.

1.1 Distributional Semantics and Computational Linguistics

Distributional semantics is a usage-based model of meaning that underlies various computational methods for semantic representation (Sahlgren 2008, Lenci 2018): it is an educational program for computers that lets them pretend they understand human languages. It relies on what is called the Distributional Hypothesis, according to which lexemes with similar meanings will have similar distributions, i.e. will occur in similar contexts. The core idea is typically attributed to Harris (1954) and Firth (1957), but exactly how enthusiastic they would be at the sight of the current implementations is disputed: Tognini-Bonelli (2001: 157) remarks that Firth would not be in favour of electronic corpora, and Geeraerts (2017) offers a comprehensive comparison between Harris' position and current distributional semantics. The attribution issue notwithstanding, the idea that meaning can be modelled by means of distributional information is pervasive in NLP and at the core of every form of Distributional Semantics. A more important question is what we mean by *meaning* or *semantics* to begin with (Sahlgren 2006, Lenci 2008), which in this research is informed by the Cognitive Linguistics framework. Beyond the particular attention to the semantic side of distributional semantics, this dissertation sets itself apart from most mainstream computational approaches in three core aspects: its motivation, the definition of units and its reliance on context-counting models.

1.1.1 Motivation

Computational Linguistics is typically task-oriented: it aims to solve concrete challenges such as information retrieval, question answering, sentiment analy-

sis, machine translation, etc. For that purpose, benchmarks or gold standards are developed and the models are tested against them. For example, Baroni, Dinu & Kruszewski (2014) test different kinds of models against datasets tailored to evaluate semantic relatedness, synonym detection, concept categorization, selectional preferences and analogy; see Agirre & Edmonds (2007) and Raganato, Camacho-Collados & Navigli (2017) for evaluation systems for sense disambiguation. This is understandable and appropriate in a task-oriented workflow: when it comes to output, it does not really matter *how* the model reached the answer, as long as it is the answer that we seek. In contrast, investigating the structure of semantic representations, i.e. the *how* of this process, calls for a different approach (see for example Baroni & Lenci 2011, Wielfaert et al. 2019). On the one hand, we do not assume that there is one correct answer because we do not assume that there is only one question. Beyond "Are these two words similar?", we are interested in: "Are they synonyms?", "Are they co-hyponyms?", "Are they regionally specific expressions of the same concept?", and so forth. Different models may focus on different dimensions of semantic structure and thus answer different questions. For that reason, the dataset collected for this research covers a wide range of semantic phenomena, in the hope of tuning distributional models to their identification. On the other hand, we are not confident that any of those questions has an unequivocal answer either. As Chapter 4 will show, annotators often agree on the sense of an utterance, but not always. Hence, the manual annotations will serve as a guideline for the interpretation of the models, but not as a law to judge their accuracy.

1.1.2 Units of analysis

Whereas computational models typically work at type-level and often with word forms, this dissertation focuses on token-level models with lemmas as units. Type-level modelling represents a lexical unit, such as *word*, as the aggregated distributional behaviour of all its occurrences, e.g. we could see that *word* tends to be preceded by *the*. Patterns can be found by accumulating and classifying contextual information from thousands if not millions of events. The profile of a type can subsequently be compared to the profiles of other types, e.g. we can see that *sentence* also tends to be preceded by *the*, while *walking* does not. Such a representation conflates the variation within the range of application of that item as part of one overall tendency, and is therefore not suited to study polysemy. Even if the context does contain disambiguating cues, such as "Can we have a *word*?", or "That *word* is not in the dictionary", the type-level representation will cover both. In spite of these shortcomings, some computational approaches to modelling polysemy do try to find the patterns in the type-level representations, e.g. Koptjevskaja-Tamm & Sahlgren (2014). In contrast, the work presented here relies on token-level modelling, which represents individual instances, e.g. comparing the two occurrences of *word* in the examples above. This approach does originate in computational linguistics (Schütze 1998) but is far less popular than type-level approaches, which are considered the default in most introductory descriptions of distributional models (Lenci 2018, Turney

& Pantel 2010, Bolognesi 2020).

Apart from the distinction between modelling types or tokens, a crucial difference between this approach and many studies in computational linguistics is that the unit of analysis is the lemma instead of the word form. On the one hand, relying on word forms avoids layers of preprocessing that already incorporate a certain interpretation in terms of what counts as a word, which different forms go together and how they are classified grammatically. Sinclair (1991) also argues along these lines for the usage of word forms as lexical units in corpus linguistics. And, admittedly, different word forms of a given lemma might exhibit diverging distributional and semantic profiles. However, from a lexicological and lexicographical perspective, centring the lemma — the combination of stems and grammatical category — is the common practice. Moreover, the mismatch between word forms and lemmas — and therefore between either of them and meanings — is highly dependent on the language we describe and the words themselves. Therefore, lemmas will be the unit of analysis in this dissertation. This is not to say that the workflow depends on this decision, in the same way that it does not depend on Dutch being the language of the corpus. The methodology presented in these pages could be applied with word forms at the centre, but the degree to which the conclusions reached here would be applicable is an empirical question.

1.1.3 Context-counting and context-predicting

Currently, the most popular approach for distributional semantics relies on neural networks, i.e. context-predicting models. The methodology followed in this project relies instead on count-based or context-counting models: the values of the vectors, i.e. numerical representations of lexical units, are (relatively) directly derived from frequency counts. In contrast, the approach initiated by Mikolov et al. (2013) and which has taken over NLP, i.e. word embeddings, is a context-predicting architecture. Neural networks are trained to predict empty slots in a fragment of text: given a fixed window with a target item in the middle, CBOW models are given the surrounding context in order to predict the target item, whereas skip-gram models try to predict the context based on the item in the middle. The training consists on a long sequence of trial and error: there is a right answer, i.e. the actual corpus, the algorithm starts by guessing and receives feedback, and iteratively it adapts its guessing strategy to minimise the error. The strategy consists of weights in the hidden layer of neural network; these weights are then used to represent the target item. In other words, while a context-counting model would define the distributional profile of a word along the lines of "it tends to co-occur with *chocolate* and *cookies* but not with *mycorrhyza* or *algorithm*", context-predicting models say, more or less, "this is how I feel/what my brain does when I see that word". The latter is, in a sense, more in line with the core of meaning as an introspective experience that defies definitions and restrictions, although computational models are far from actually *understanding* language. Exploring to what degree these models approximate humans' assessments lies in the purview of other research programmes involving psycholinguistic experiments. Studies have been carried

out to compare the performance of context-counting and context-predicting models — in terms, of course, of their accuracy with regards to popular benchmarks. Baroni, Dinu & Kruszewski (2014) found that the word2vec architecture outperformed context-counting models, much to their disappointment. In contrast, Levy, Goldberg & Dagan (2015) fine-tuned context-counting models based on the hyperparameters from word embedding and found that performance differences where local or even insignificant.

When our purpose is to understand what of meaning, if anything, can be found in text data, the interpretation of context-counting models is much more transparent. We can trace the composition of the vectors to concrete frequencies and instances. As we will see in the second part of this dissertation, these supposedly more transparent models are already quite opaque, especially with the added transformation from type-level to token-level models. That said, most of the workflow described here can also be combined with context-predicting models.

The years since Mikolov et al. (2013) have seen a rapid and enthusiastic growth in the field of word embeddings and NLP, with new models continually surpassing the previous ones. One of these is BERT (Devlin et al. 2019), which, in spite of its indubitable relevance to the approach proposed here, will not be explored. Bidirectional Encoder Representations from Transformers (BERT) is a machine-learning technique that can represent individual instances and sentences: unlike other context-predicting models, it can be used for token-level representations. But like other context-predicting models, its output is somewhat less interpretable than context-counting models. It has been tested on the typical task-based benchmarks and it is so time- and resources-consuming that NLP researchers will typically use pre-trained embeddings and fine-tune them for specific tasks rather than generate them from scratch. In principle, combining a model of the BERT family with the workflow described here is not impossible: as long as occurrences are represented with vectors from which we can derive pairwise distances, the rest of the analysis stays the same. However, some crucial differences remain: we do not know which elements of the context informed the models' decision, they are based on word forms and the word forms are based on a different tokenizer. For instance, a brief test of BERTje (de Vries et al. 2019), the Dutch counterpart of BERT, on a section of the dataset used for this project revealed that (i) for some lemmas BERTje's answer might be closer to the human perspective, (ii) for other lemmas a deeper investigation is in order and (iii) other lemmas cannot be modelled at all because of the discrepancy in the tokenization procedure[1]. In other words, even if combining the methodologies is possible, the actual implementation requires some planning, specific decisions and tailoring the procedure to extract as much as we can from the backstage operations in context-predicting models.

[1]The comparison was applied to a few lemmas, including *hoop* 'hope/heap', *dof* 'dull' and *heilzaam* 'healthy/beneficial'. In the first case, which was particularly challenging for the context-counting models, BERTje outperformed them; in the second, some context-counting models outperformed BERTje; and the third was never identified as one unit by BERTje's tokenizer.

1.2 Distributional Semantics and Cognitive Semantics

As a computational approach, distributional semantics is not intrinsically linked to any particular linguistic theory. Its usage-based essence makes it a natural fit for approaches that describe the *parole* along with the *langue* (in terms of de Saussure 1971), such as Cognitive Linguistics. In the introduction to *The Oxford Handbook of Cognitive Linguistics*, it is described as

> an approach to the analysis of natural language that originated in the late seventies and early eighties in the work of George Lakoff, Ron Langacker, and Len Talmy, and that focuses on language as an instrument for organizing, processing, and conveying information. (Geeraerts & Cuyckens 2007b: 3)

It stands in contrast to frameworks that uphold a strict separation of semantics and pragmatics, of structure and usage, of lexical knowledge and world knowledge (Geeraerts 2010b). As the introduction and composition of the *Handbook* shows, as well as other compilations along these lines (such as Rudzka-Ostyn 1988, Kristiansen et al. 2006, Ibarretxe-Antuñano & Valenzuela 2016), the diverse field of Cognitive Linguistics is guided by a number of principles derived from this central notion of language as categorization. Among these principles, three in particular constitute the theoretical cornerstones of this study: (i) an emphasis on meaning, (ii) the notion of fuzzy and prototypical categories and (iii) a usage-based approach.

1.2.1 Everything is semantics

Understanding language as categorization and its function in the organization and communication of knowledge necessarily places the focus on meaning (Geeraerts & Cuyckens 2007a, Geeraerts 2016). From a Cognitive Linguistics perspective, all linguistic structures — not just lexical items but also syntactic patterns — are considered inherently meaningful (Langacker 2008, Lemmens 2015). Moreover, meaning in Cognitive Linguistics goes beyond traditional semantics — i.e. distinguishing linguistic from nonlinguistic features — and includes encyclopedic knowledge and pragmatics (Glynn 2010, Geeraerts 1997). While it is crucially a cognitive phenomenon involving conceptualization, it takes place in the mind of physical, embodied beings who perceive, understand, and interact with their world: meaning is embodied and neither limited to nor separated from reference (Rohrer 2007).

The centrality of semantics in Cognitive Linguistics has led to a strong body of work on meaning and on how traditional notions fit in with cognitive principles. For example, the line of work initiated in the '80s with Lakoff & Johnson (2003) and further developed along different lines by Raymond Gibbs Jr., Gerard Steen, Zoltán Kövecses, Elena Semino and many others (see for example Gibbs & Steen 1999, Gibbs Jr. 2008, Semino 2008, Kövecses 2015) builds on understanding a traditional linguistic concept, i.e. metaphor, with the tools of Cognitive Linguistics. In these terms, metaphor refers to ways of

thinking, understanding, conceptualizing, that manifest in linguistic behaviour but also permeate other areas of everyday life.

Along these lines, relationships between senses are understood as cognitive mechanisms that need not be restricted to linguistic behaviour nor to extralinguistic reference. Semantic categories such as metaphor, metonymy, specialization, homonymy and prototypicality are crucial tools to make sense of the variety of relationships between what we understand as senses. They are not unique to Cognitive Linguistics, but a framework that understands meaning as a property of any linguistic structure and as covering linguistic and extralinguistic features allows us to look for meaning in distributional models without expecting them to exhaust semantic description.

Cognitive Linguistics also incorporates the combination of a semasiological and onomasiological perspective, while previous frameworks have defined either one or the other as the only possibility (Geeraerts 2010b). A semasiological perspective, which is predominant in the research described here, starts from a form or expression and investigates its range of meanings or applications, e.g. the study of polysemy. An onomasiological perspective, on the other hand, starts from a concept and describes the forms that are used to express it, e.g. synonymy. This dissertation takes a semasiological perspective, but token-level distributional models can be used from both perspectives, as shown in De Pascale (2019).

1.2.2 Prototypicality

Among the most important notions in the Cognitive Linguistics understanding of categorization we find prototypicality and salience (Rosch 1978). Categories cannot always be described in terms of necessary and sufficient conditions; instead, they may be characterized by clusters of co-occurring properties that do not apply to all members to the same degree. They may even have fuzzy boundaries, an unclear range of application. As a property of categorization, this is a property of language, which Cognitive Linguistics embraces, incorporating a quantitative dimension to the study of meaning (Geeraerts 2010b). At this point, a quantitative perspective does not immediately require statistical methods, but refers to a shift in the understanding of what counts as meaning description. The notion of prototypicality makes it interesting, if not inevitable, to look at the uneven distribution and importance of the different features or members of a category, as is done, for example, in Geeraerts, Grondelaers & Bakema (1994) and Geeraerts (1997):

> …the essence of prototype theory lies in the fact that it highlights the importance of flexibility (absence of clear demarcational boundaries) and salience (differences of structural weight) in the semantic structure of linguistic categories. (Geeraerts 2006: 74)

Given the set of meanings that a form can express, i.e. the intensional level, some of them are more salient than others. For example, given my current lifestyle, 'device to control the cursor on a screen' is a more salient meaning of *mouse* than 'small rodent'; but, crucially, this might not be the case in

other contexts, for other speakers. Given the range of application of a form or a meaning (i.e. the extensional level), some may be more typical members than others. For instance, a black, minimalist computer mouse might be more typical than a wavy, wider gaming mouse with a bright green drawing of a dragon. These situations represent intensional and extensional nonequality, respectively: some senses or members of a category are better representatives of the category than others. Both dimensions may overlap: a typical computer mouse concentrates most of the typical features of the category, regarding its functionality, size, shape and colour; conversely, a typical feature is defined by occurring frequently in the members of the category. These are two of the characteristics of prototypicality, and are complemented by intensional and extensional non discreteness, i.e. the lack of a single set of necessary and sufficient conditions and fuzzy boundaries of the categories. As could be expected, even prototypicality is a prototypical category, as these four features need not co-exist. The relative salience of the two senses of *mouse* does not mean that we might find an unknown entity and be in doubt whether it is a mouse; meanwhile, discussions such as whether a tomato is a fruit might easily ensue. Geeraerts (2006: Ch. 4) offers a typology of salience phenomena as an application of prototype theory beyond the semasiological structure. For example, if from the semasiological perspective we are interested in describing how frequent (or salient) apples are as referents for the word *fruit*, from the onomasiological perspective we are interested in how frequently the word *fruit* is used to refer to apples (compared to saying *apple*).

The notion of (semasiological) prototypicality will be relevant for the interpretation of the modelling in Chapter 6. Until then, it also permeates the understanding of meaning that underlies this research. On the one hand, fuzzy boundaries and degrees of membership invite us to rethink the usefulness of reified senses: ambiguous examples and overlapping features are to be expected. Instead, a bottom-up procedure would rather capture configurations of features (Glynn 2014); assigning discrete senses to corpus data imposes a categorical structure that we know to be inappropriate (see also Geeraerts 1993). On the other hand, distributional models, as a quantitative approach that measures similarity between entities, is particularly adequate to such a non-discrete representation.

In this dissertation I will continue to talk about senses and I will extract discrete patterns from the non-discrete representations in terms of clusters, in order to manipulate and talk about these abstract entities, without implying that they have any ontological reality beyond the explanatory purposes. When it comes to senses, they are not considered a gold standard, an unique solution to the semasiological description of a lexical item; instead, they are guides and an operationalization of certain research questions. The clusters, on the other hand, will be generated by an algorithm that is forced to produce discrete groups but does assign its elements different degrees of membership (see Section 2.2.4). Finally, the overall approach describes tendencies, preferences, probabilities: at no level are the categories and typologies offered in this dissertation discrete and uniform. I have tried, but language resists.

1.2.3 A usage-based approach

Cognitive Linguistics presents itself as a usage-based approach and, as such, it is entirely compatible with a bottom-up, empirical, quantitative methodology such as distributional semantics. Quantitative cognitive semantics is now an established field, as shown by the contributions gathered in Gries & Stefanowitsch (2006), Glynn & Fischer (2010) and Glynn & Robinson (2014), among others. However, not all of Cognitive Linguistics — and especially Cognitive Semantics — relies on empirical methods: introspection was still the main source of information in much of the foundational sources (see for example the discussion illustrated in Geeraerts 1999). In practice, both introspection and empirical methods are required in scientific research, albeit applied to different stages or aspects of the investigation (Geeraerts 2010a). Interpretation is needed in order to formulate hypotheses that will guide the data collection and analysis and to interpret the results: the data does not speak for itself. The empirical steps, in contrast, facilitate reproducibility and falsifiability: by describing the concrete corpus, the method of collection and the quantitative methods applied to it, the study can be replicated by different researchers and the results compared. At the same time, large-scale quantitative methods such as distributional semantics delegate time consuming or computationally expensive tasks, such as reading and comparing thousands of attestations of a word, to an automatic system that can perform it faster and more systematically than humans, leaving the researcher to dedicate their energies in the tasks that humans are best at: interpretation and creativity. That is precisely the long-term goal of this research: to offer an empirical, quantitative workflow that transforms huge amounts of data, finds relevant patterns and provides them to the linguist for interpretation and the formulation of hypotheses.

Empirical research in semantics can take different shapes: corpus-based methods, as is the case in this research, but also experimental and referential methods. As Geeraerts (2015: 242-243) argues, each of these approaches captures a different aspect of meaning, namely textual patterns, on-line processing or referential properties. Meaning, especially from the maximalist perspective taken in Cognitive Linguistics, is too complex to be fully described by any one of these methods in isolation (see also Arppe et al. 2010, Stefanowitsch 2010). As such, we do not have such high expectations from distributional semantics — part of the question is: *what* do these models say? Concretely, we do not expect distributional models to provide information on how we *think*, but on how a community speaks and categorises: " 'language as cognition' encompasses shared and socially distributed knowledge and not just individual ideas and experiences" (Geeraerts 2016: 533). It is the pool of shared practices and knowledge that corpora offers and distributional semantics tries to model.

Moreover, despite the large corpora, the advanced quantitative techniques and the sophisticated visualization tools on which this dissertation is built, this study has its limits. It is restricted to a specific corpus, and as such to specific varieties of a specific language, to a specific genre and period in time, to written text; it is restricted to a limited set of lexical items that were investigated; it is restricted to the precise samples collected, the precise questions asked, the precise techniques used to answer them. Most importantly, I will be as thorough

as possible in stating the conditions in which the research was carried out and the choices made along the way. As a result, these limits are not just warnings as to the range of applicability of the results and conclusions, but also and more importantly sources of possibilities, inspiration for similar studies facilitated by the empirical nature of the investigation.

1.3 Visual analytics

Distributional models return mathematical representations of lexical items — or, in the case of token-level models, their attestations. These mathematical representations are arrays of numbers that, in the best-case scenario, we can interpret as co-occurrence information, as an unsorted list of collocations. We need an additional step to transform these individual representations into similarities, which operationalize the Distributional Hypothesis mentioned above. However, even then, the output is a matrix with as many rows and columns as items we are comparing; depending on the magnitude of our sample and the subtlety of its structure, scanning it visually can be taxing, if not entirely in vain. So, that is not what we do.

For word sense disambiguation, evaluation would normally involve a clustering algorithm, a benchmark and a measure of accuracy. The clustering algorithm would take the vectors or the similarity matrix and return clusters: groups of similar items that are different from each other. The measure of accuracy would report on the agreement between the clustering solution and the benchmark: the closer they are, the better the model. However, these measures say nothing about the qualitative differences between models, i.e. whether they misclassified the same items or how they differ from the benchmark. Even if we take the gold standard as an actual ground truth and the only correct solution — which is not the case in this study — this is not an ideal situation.

It is responding to these concerns that a visualization tool for the exploration of token-level models was envisaged (Wielfaert et al. 2019). The tool developed by Wielfaert in the context of the Nephological Semantics project takes the output from a dimensionality reduction algorithm, i.e. a procedure that tries to map distances based on multiple dimensions on a 2D or 3D space, and surrounds its visual representation with interactive features. These additional features, tailored for the exploration of distributional models, set the tool apart from a static scatterplot, or even from a default interactive plot.

To put it in Card, Mackinlay & Shneiderman (1999: 6)'s words: " 'The purpose of visualization is insight, not pictures'. The main goals of this insight are *discovery*, *decision making* and *explanation*"[2]. Indeed, the kind of qualitative exploration achieved through this tool would have been extremely hard without it, if not impossible. In the first place, the tool sets up a workflow that goes from the exploration of the similarity *between* models and the role of parameter settings through the qualitative comparison of selections models to the detailed exploration of individual models. It is built to facilitate a fluid exploration and interconnection between levels of analysis. The tool offers simultaneous, in-

[2]Emphasis from the original.

terconnected access to the actual output of a model (as coordinates on a 2D plane), the variation of parameter settings, semantic annotation, metadata of the corpora and frequency data on the context words. The interaction of these different aspects of distributional models in a practical visual interface makes patterns and insights accessible that would not have been found any other way.

Because of this, the visualization tool is a key component of this dissertation. It is in these scatterplots that we find the clouds: clusters of similar tokens that come together in denser areas of the (reduced) semantic space. In an actual case study involving the methodological workflow presented here, a lot of the technicalities go into generating the clouds, but a large part of the analysis involves looking at them and finding shapes: cloudspotting.

1.4 Nephological Semantics

The research presented in this dissertation is part of a larger project within the QLVL research unit, the BOF C1-project (3H150305) "Nephological Semantics: using token clouds for meaning detection in variationist linguistics", with Dr. Prof. Dirk Geeraerts as Principal Investigator. Both the Python module for the creation of the models, written by Tao Chen, and the visualization tools for their analysis, designed by Thomas Wielfaert and myself, are products of this project. Moreover, this dissertation would not be what it is without the integration of the case studies, questions and insights discussed here with other branches of the project, and without the feedback loop on ideas, tests and thoughts on the different techniques.

The main objective of the project is to develop — and understand — appropriate methods for the retrieval of semantic information from corpus data, addressing concerns that stem from a longer tradition of usage-based lexical research. Geeraerts, Grondelaers & Bakema (1994) and Geeraerts, Grondelaers & Speelman (1999) embark in comprehensive, detailed lexicological analyses of the lexical fields of clothing and football terms in Dutch. Their approach is referential: Geeraerts, Grondelaers & Bakema (1994), for instance, collect pictures and descriptions of garments from Dutch and Flemish magazines and describe each clothing item in terms of a variety of features, such as the length of the sleeve. Based on the relationship between the (configurations of) features and the items used to name the objects, they developed a model of lexical variation that takes into account prototypicality and salience in terms of semasiological, onomasiological and contextual variation. However, the manual and detailed identification of features at a large enough scale is painstaking and time consuming, if at all feasible. In contrast, machine-readable linguistic material is available, more or less accessible and, given the right resources, processable. It will not provide the same kind of information as a referential approach, but it is more easily scalable to large amounts of data.

In the context of this project, token-level models for semasiological research are introduced by Heylen, Speelman & Geeraerts (2012) and Heylen et al. (2015). Another work-package, culminating in De Pascale (2019)'s PhD dissertation, applies the technique to lexical lectometric research, i.e. measuring distances between language varieties based on their naming choices for differ-

ent concepts. The visualization tool, as mentioned before, is first described in Wielfaert et al. (2019). Between their work, this dissertation and further case studies taking place in the last year, the project is covering the application of token-level vector space models on semasiological, onomasiological and lectometric studies in varieties of Dutch and Mandarin, at both a synchronic and a diachronic level.

1.5 Structure of the dissertation

As a product of the Nephological Semantics project, this dissertation aims to contribute to both the development and understanding of distributional models for lexical semasiological research. It brings together the theoretical perspective on semantics from Cognitive Linguistics with computational methods and visual analytics in the hope of paving the way for future research along the same lines. With that in mind, the three chapters of the first part of this dissertation, *The cloudspotter's toolkit*, will focus on the technical or methodological side of the project. Chapter 2 will describe the procedure to create clouds and the parameter settings explored, taking care to be thorough and specific about the technical decisions that resulted in the final models. Then, Chapter 3 will showcase the visualization tool designed by Thomas Wielfaert and myself as well as a ShinyApp extension that provides additional functionalities. Finally, Chapter 4 will illustrate the dataset on which the models were tested: the selection of lemmas and the questions they try to address, the collection of data and the annotation procedure.

The notion behind token-level models, i.e. that we can represent meaning differences in terms of distributional differences, and in particular the image of a scatterplot that translates these intuitions into an interpretable picture, *sounds good*. Alas, the reality is not as bright as we could have wished for, and the skies of distributional semantics have all but a stable weather. Hopefully, this dissertation can offer a guide for researchers who would dare to tread these waters. Therefore, the three chapters in the second part, *The cloudspotter's handbook*, will discuss the results of the analyses, with an emphasis on crucial assumptions that clash with the data. First, Chapter 5 debunks the idea of a perfect cloud emerging from the ocean of the corpus. Clouds come in many different shapes, caused by different phenomena of distributional behaviour, and thus this chapter offers a classification of what we might encounter. Chapter 6 follows with a linguistic perspective on the variation of these shapes and discusses what we can or we cannot find in these models. Finally, Chapter 7 shows how no set of parameter settings offers the best solution across the board — not even close. Instead, the same parameter settings may result in different shapes for different lemmas, and they have to be tailored *to the specific lemma* to capture the relevant semantic structure.

An enthusiastic and hopeful aspiring cloudspotter might feel discouraged by the variability — bordering on unpredictability — of these clouds. I wouldn't blame them. However, in spite of the diversity of shapes, of semantic phenomena and of parameter settings to explore, the methodology can offer interesting insights. They are partial insights, but insights nonetheless, and once we know

what to expect from clouds, we can focus on acquiring them. In that perspective, the third and final part of this dissertation, *The cloudspotter's cheatsheet*, will close with a general practical guide, a summary of suggestions for further research and an overall conclusion.

Part I

The cloudspotter's toolkit

Chapter 2

From corpora to clouds

The main goal of the methodological framework presented here is to explore semasiological structure from textual data. The starting point is a corpus, i.e. a selection of texts, and one of the most tangible outputs is what we will call *clouds*: the visual representation of textual patterns as dense areas in a 2D scatterplot. In this chapter we will explain how to generate clouds from the raw, seemingly indomitable ocean of a corpus.

First, we will describe how token-level vector space models are created: these are mathematical representations of the occurrences of a lexical item. We will focus on context-counting models, but this is by no means the only viable path. Other techniques, such as BERT (Devlin et al. 2019)[1], that can also generate vectors for individual instances of a word, could be used for the first stage of this workflow. Section 2.1 will describe the process and the rationale without assuming a strong mathematical background for the reader, leaving the deeper technicalities to Section 2.2. In Section 2.3, we will break apart the workflow into the multiple choices that the researcher needs to make and that result in a potentially infinite number of models, while Section 2.4 briefly presents a method to select a few representative models. Finally, Section 2.5 summarizes the chapter.

2.1 A cloud machine

At the core of vector space models, *aka* distributional models, we find the Distributional Hypothesis, which is often linked to Harris's observation that "difference of meaning correlates with difference of distribution" (1954: 156), but also to Firth's "You shall know a word by the company it keeps" (1957: 11) and Wittgenstein's "the meaning of a word is its use in the language"[2] (1958: 20). In other words, items that occur in similar contexts in a given corpus will be semantically similar, while those that occur in different contexts

[1] See also de Vries et al. (2019) for a Dutch version.

[2] The famous quote is preceded by an appropriate nuance: "For a *large* class of cases — though not for all — in which we employ the word 'meaning' it can be defined thus: the meaning of a word is its use in the language".

will be semantically different (Jurafsky & Martin 2020: Ch. 6, Lenci 2018). Crucially, this does not imply that we can describe an individual item with their distributional properties, but that comparing the distribution of two items can tell us something about their semantic relatedness (Sahlgren 2006: 19).

Firth (1957) inspired generations of corpus linguists to look at collocations as part of the semantic description of a lemma. The Birmingham school, pioneered by John Sinclair, used co-occurrence frequency information to describe a lexical item by the set of those context words most attracted to them. Due to the skewed distribution of word frequencies, known as Zipf's law, this attraction cannot be measured in terms of raw co-occurrence frequencies. For example, the most frequent lemma in the (Dutch) corpus used for this research, discarding punctuation, is *de* 'the (fem./masc.)', which occurs 28.1 million times. The second most frequent lemma, *van* 'from', occurs 12.6 milion times, and it is followed by *het* 'the (neutral)' and *een* 'a, an', with corresponding frequencies of 11.7 and 11.1 million times each. For every 100 words in the corpus, excluding punctuation, 14 are one of these four words. Of the total of 4.6 million different words, 61% are *hapax legomena*, i.e. they occur once, and 172 lemmas cover 50% of all the occurrences. As a consequence, co-occurrences with very frequent words are not as informative as those with less frequent words, and hence raw co-occurrence frequencies are transformed to measures of **association strength**, such as mutual information (see Section 2.2.1) or t-score, among others (for an overview see Evert 2009, Gablasova, Brezina & McEnery 2017). In collocational studies, researchers typically set a threshold of association strength and only look at the context words that surpass it.

At their core, context-counting **vectors** are lists of association strength values. Each word is represented by its association strength to a long array of words that it might co-occur with, as shown in Table 2.1. Unlike in collocation studies, low values — or even lack of co-occurrence — are not excluded, but used in the comparison with other words that might. Going back to the Firthian motto, a collocational study would describe me with the list of people that I talk to the most, whereas a distributional model would compare me to someone else based on who either of us talks to and how often we talk to them. The more people we have in common, the more similar we are, but people that neither of us talks to have no impact on the comparison.

Table 2.1 shows small vectors representing the English nouns *linguistics*, *lexicography*, *research* and *chocolate*, as well as the adjective *computational*, with co-occurrence information obtained from the GloWbE (Global Word-based English) corpus. The values are their association strength PMI with each of the lemmas in the columns: the higher the values, the stronger the attraction between the word in the row and the word in the column (See Section 2.2.1). From a collocational perspective, *linguistics* is strongly attracted to both *language* and *English*, i.e. they occur very often in a span of 10 words from each other, considering their individual frequencies; it is less attracted to *word* and *to speak*, and does not co-occur with either *to eat* or *Flemish* within that window, in this corpus.

Each row in Table 2.1 is a vector coding the distributional information of the lemma it represents. By **lemma** we refer to the combination of a stem and

Table 2.1: Small example of type-level vectors, with PMI values based on a symmetric window of 10. Frequency data extracted from GloWbE.

target	language/n	word/n	flemish/j	english/j	eat/v	speak/v
linguistics/n	4.37	0.99	-	3.16	-	0.41
lexicography/n	3.51	2.18	-	2.19	-	2.09
computational/j	1.6	0.08	-	-1	-	-1.8
research/n	0.2	-0.84	0.04	-0.5	-0.68	-0.38
chocolate/n	-1.72	-0.53	1.28	-0.73	3.08	-1.13

Note:
Part-of-speech is indicated after a slash: n = noun, j = adjective, v = verb

a part of speech, e.g. *chocolate/n* covers *chocolate*, *chocolates*, *Chocolate*, etc. These vectors are meant to code the distributional behaviour of the linguistic forms they represent — in this case lemmas —, in order to operationalize the notion of distributional similarity and, consequently, model their meaning. For example, in Table 2.1 the first two rows, representing *linguistics* and *lexicography*, are similar to each other: both words have a similar attraction to *language* and to *English*, even if the values for *word* and *to speak* are more different. More importantly, they are more similar to each other than to other rows in the table, which have lower values for those four columns and might even co-occur with *Flemish* and *to eat* as well. The Distributional Hypothesis expresses the observation that words that are distributionally similar, like *linguistics* and *lexicography*, are semantically similar or related, whereas words that are distributionally different, like *linguistics* and *chocolate*, are semantically different or unrelated.

The rows in this table are **type-level vectors**: each of them aggregates over all the attestations of a given lemma in a given corpus to build an overall profile. As a result, it collapses the internal variation of the lemma, i.e. its different senses or semasiological structure. In order to uncover such information, we need to build vectors for the individual instances or **tokens**, relying on the same principle: items occurring in similar contexts will be semantically similar. For instance, we might want to model the three (artificial) occurrences of *study* in (1) through (3), where the target item is in bold and some context words are in italics.

(1) Would you like to **study** *lexicography*?
(2) They **study** this in *computational linguistics* as well.
(3) I eat *chocolate* while I **study**.

Given that, at the aggregate level, a word can co-occur with thousands of different words, type-level vectors can include thousands of values. In contrast, token-level vectors can only have as many nonzero values as the individual window size comprises, which drastically reduces the chances of overlap between vectors. In fact, the three examples don't share any item other than the target. As a solution, inspired by Schütze (1998), (a selection of) the context words around the token is replaced with their respective type-level vectors (Heylen,

Table 2.2: Small example of token-level vectors of three artificial instances of *to study*.

target	language/n	word/n	english/j	speak/v	flemish/j	eat/v
$study_1$	4.37	0.99	3.16	0.41	0.00	0.00
$study_2$	5.97	1.07	2.16	0.00	0.00	0.00
$study_3$	0.00	0.00	0.00	0.00	1.28	3.08

Speelman & Geeraerts 2012, Heylen et al. 2015, De Pascale 2019). Concretely, example (1) is represented by the vector for its context word *lexicography*, that is, the second row in Table 2.1; example (2) by the sum of the vectors for *linguistics* (row 1) and *computational* (row 3); and example (3) by the vector for *chocolate* (row 5). This not only solves the sparsity issue, ensuring overlap between the vectors, but also allows us to find similarity between (1) and (2) based on the similarity between the vectors for *lexicography* and *linguistics*. As we will see in Section 2.3, we can even use the association strength between the context words and the target type, i.e. *to study*, and give more weight to the context words that are more characteristic of the lemma we try to model. The result of this procedure is a co-occurrence matrix like the one shown in Table 2.2. Each row represents an instance of the target lemma, e.g. *to study*, and each column, a lemma occurring in the corpus[3]; the values are the (sum of the) association strength between the words that occur around the token, i.e. their **first-order context words**, and each of the words in the columns, i.e. the **second-order context words**. In addition, all negative and missing values value have been set to zero, due to the unreliability of negative PMI values (see Section 2.2.1).

The next step in the workflow is to compare the items to each other. We can achieve this by computing cosine distances between the vectors (see Section 2.2.2 for the technical description). The resulting distance matrix, shown in Table 2.3, tells us how different each token is to itself, which takes the minimum value of 0, and to each of the other tokens, with a maximum value of 1. We can see that (1) and (2) are very similar to each other, because they co-occur with similar context words, i.e. *linguistics* and *lexicography*, but drastically different from (3), which was modelled based on *chocolate*. The specific selection of context words is crucial: if we had selected *computational* but not *lexicography* to model (2), it would have resulted in a larger difference with (1). The series of choices that we can make and that have been made for this research project are discussed in Section 2.3.

Table 2.3 is small and simple, but what if we had hundreds of tokens? The more items we compare to one another, the larger and more complex the distance matrix becomes. In order to interpret it, we need more stages of processing. On the one hand, dimensionality reduction techniques such as

[3]Which lemmas in particular are a matter for Section 2.3.4, but in any case, lemmas that do not co-occur with any of the context words of the tokens will have zeros in all the rows and therefore be dropped.

Table 2.3: Cosine distance matrix between the three artificial instances of *to study*.

	study_1	study_2	study_3
study_1	0.00	0.04	1
study_2	0.04	0.00	1
study_3	1.00	1.00	0

MDS, t-SNE and UMAP, which will be discussed in Section 2.2.3, offer us a way of visualizing the distances between all the models by projecting them to a 2D space. We can then represent each model into a scatterplot, like in the plots of Figure 2.1, where each point represents a token, and their distances in 2D space approximate their distances in the multidimensional space of the co-occurrence matrix. Visual analytics, such as the tool described in Chapter 3, can then help us explore the scatterplot to figure out how tokens are distributed in space, why they form the groups they form, etc.

Word Sense Disambiguation makes use of clustering algorithms to extract clusters of similar tokens from their models. The idea behind it is that, if distributional similarity correlates with semantic similarity, groups of similar tokens should share the same sense and have a different sense from other groups of tokens. In Chapter 6 we will see to what degree this assumption holds in this data and with these methods.

The final step in our workflow is, then, the combination of dimensionality reduction and clustering, which results in the right plot of Figure 2.1: by means of dimensionality reduction, tokens are located in a scatterplot so that distributional similarities are approximated as spatial similarities, and groups of similar tokens are assigned different colours. In previous research, which did not integrate clustering procedures in this manner, the term *cloud* was used to refer to a full model (Heylen et al. 2015, Wielfaert et al. 2019, De Pascale 2019, Montes & Heylen Submitted). In this study, instead, *cloud* will refer to each of the clusters, identified by colours in the scatterplot.

2.2 The chemistry of cloud making

A typical vector space model is an item-by-feature matrix: its rows code items, its columns code features, and its cells code information related to the frequency with which the items and features co-occur. The first distributional models counted the occurrences of words in documents and represented them in word-by-document matrices; the models described here are token-by-feature matrices, in which the rows are attestations of a lexical item and the features are second-order co-occurrences, i.e. context words of the context words of the token. Turney & Pantel (2010) offer an overview of different kinds of matrices, based on the items modelled and the features used to describe them. Besides matrices, vector space models can be tensors, which are generalisations of matrices for more dimensions and can allow for more complex interactions,

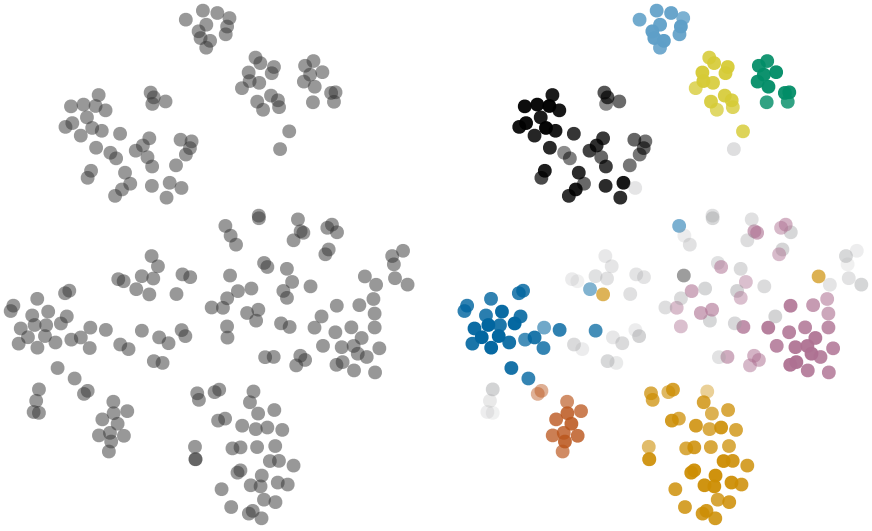

Figure 2.1: 2D representation of Dutch *hachelijk* 'dangerous/critical'.

e.g. subject-verb-object triples in Van de Cruys, Poibeau & Korhonen (2013); see also Lenci (2018).

Models can be based on co-occurrence counts, as is the case in these studies, or on machine-learning algorithms trained to predict the context around a word or fill in an empty slot given a few words around it. These context-predicting models use the weights of their neural networks as features in the vectorial representations of the words they predict. A number of papers have explored which kind of models work best for different tasks, with uncertain results (Baroni, Dinu & Kruszewski 2014, Levy, Goldberg & Dagan 2015). As explained before, such context-predicting models will not be explored in this dissertation, although their integration would be interesting for further avenues of research.

The workflow described in the previous section relies on mathematical principles to obtain linguistic patterns from a (mostly) raw corpus. A full understanding of the formulae that underlie each step is not necessary to grasp the gist of this methodology, but it is required for an appropriate implementation. In this section, we will take a deeper look into the technical aspects the machinery behind the process, in particular association strengths, similarity metrics, dimensionality reduction techniques and clustering algorithms.

2.2.1 Association strength: PMI

The distribution of words in a corpus follows a power law: a few items are extremely frequent, and most of the items are extremely infrequent. Association measures transform raw frequency information to measure the attraction be-

tween two items while taking into account the relative frequencies with which they occur. They typically manipulate, in different ways, the frequency of the node $f(n)$, the frequency of its collocate $f(c)$, their frequency of co-occurrence $f(n, c)$ and the size of the corpus N. Evert (2009) and Gablasova, Brezina & McEnery (2017) offer an overview of how different measures are computed and used in corpus linguistics; Kiela & Clark (2014) compare measures used in distributional models.

In the studies discussed here, I will only use **(positive) pointwise mutual information**, or (P)PMI (Church & Hanks 1989), one of the most popular measures both in collocation studies and distributional semantics (Bullinaria & Levy 2007, Kiela & Clark 2014, Jurafsky & Martin 2020, Lapesa & Evert 2014). Its formula is shown in equation (2.1), where $p(n) = \frac{f(n)}{N}$, i.e. the proportion of occurrences in the corpus that correspond to n.

$$I(n, c) = \log \frac{p(n, c)}{p(n)p(c)} = \log \left(\frac{f(n, c)}{f(n)f(c)} N \right) \tag{2.1}$$

Negative PMI values tend to be unreliable, so positive PMI or PPMI is used, in which the negative PMI values are turned to zeros (Bullinaria & Levy 2007, Kiela & Clark 2014, De Pascale 2019, Jurafsky & Martin 2020: 109). Furthermore, PMI is known for its bias towards infrequent events: when either $p(n)$ or $p(c)$ is very low, PMI tends to be very high. In collocation studies, this bias may be counteracted by combining PMI filters with other measures that favour frequent co-occurrences, such as t-scores or log-likelihood ratio (McEnery, Xiao & Tono 2010). In distributional semantics, the accuracy of models that rely on PPMI seems not to be affected by the issue presented by this bias; moreover, in these studies any lemma with $f(n) < 217$, i.e. occurring less than once every two million tokens, was excluded, to avoid too sparse, uninformative vectors.

2.2.2 Similarities and distances: cosine

After obtaining the token-by-feature matrices, the distances between the vectors must be computed. Typically, the implementations for the dimensionality reduction and clustering can take the item-by-feature matrices as input and compute the distances under-the-hood, but they do not necessarily offer the option of computing our distance measure of choice, cosine.

Cosine is a measure of similarity between vectors $\mathbf{v}$ and $\mathbf{w}$ and is defined in equation (2.2); it coincides with the normalised dot product of the vectors (Jurafsky & Martin 2020: 105).

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}|\,|\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}} \tag{2.2}$$

For positive values, e.g. when using PPMI, the cosine similarity ranges between 0 and 1: it will be 1 between identical vectors and 0 for orthogonal

vectors, which do not share nonzero dimensions, like study_1 and study_3 in Table 2.2. Cosine is sensitive to the angle between the vectors, and not to their magnitude: the similarity between study_1 and a vector created by multiplying all the cells in study_1 by any constant will still be 1.

Cosine similarity is the most common metric in distributional models (Jurafsky & Martin 2020: 105) and has been shown to outperform other measures, especially when combined with PPMI (Kiela & Clark 2014, Lapesa & Evert 2014, Bullinaria & Levy 2007)[4]. One of the ways in which it is used is for semantic similarity tasks: the nearest neighbours of an item are extracted, by selecting the vectors with highest cosine similarity to the target vector. In these studies, similarities are usually transformed to distances by inverting the scale ($\text{cosine}_{\text{dist}} = 1 - \text{cosine}_{\text{sim}}$), so that identical vectors — and each vector to itself — have a cosine distance of 0 and orthogonal vectors have a cosine distance of 1, as shown in Table 2.3.

Before applying dimensionality reduction or clustering algorithms, the cosine distances were further transformed with the aim of giving more weight to short distances, i.e. nearest neighbours, and decreasing the impact of long distances. For each token vector $\mathbf{v}$ with n dimensions, we define the transformed vector $\mathbf{v}_{\text{transformed}}$ as $\mathbf{v}_{\text{transformed}_i} = \log(1 + \log rank(\mathbf{v})_i)$ for each i, with $1 \leq i \leq n$, and where $rank(\mathbf{v})_i$ is the similarity rank of the ith value in $\mathbf{v}$. For example, if originally we have the distances $\mathbf{v} = [0, 0.2, 0.8, 0.3]$, the rank transformation returns $rank(\mathbf{v}) = [1, 2, 4, 3]$, which after the first logarithm transformation becomes $[0, 0.693, 1.39, 1.099]$ and, after the second transformation, $\mathbf{v}_{\text{transformed}} = [0, 0.52, 0.86, 0.74]$. On the one hand, the magnitude of the distance is not as important as its ranking among the nearest neighbours. On the other, the lower the ranking, the smaller the impact: the difference between the final values for ranks 1 and 2 is larger than between ranks 2 and 3. The new matrix, where each row $\mathbf{v}$ has been replaced with its $\mathbf{v}_{\text{transformed}}$, is converted to euclidean distances.

While cosine distances are used to measure the similarity between token-level vectors, euclidean distances will be used to compare two vectors of the same token across models, and thus compare models to each other. Concretely, let's say we have two matrices, $\mathbf{A}$ and $\mathbf{B}$, which are two models of the same sample of tokens, built with different parameter settings, and we want to know how similar they are to each other, i.e. how much of a difference those parameter settings make. Their values are already transformed cosine distances. A given token i has a vector $\mathbf{a}_i$ in matrix $\mathbf{A}$ and a vector $\mathbf{b}_i$ in matrix $\mathbf{B}$. For example, i could be example (2) above, and its vector in $\mathbf{A}$ is based on the co-occurrence with *computational* and *linguistics*, as shown in Table 2.2, while its vector in $\mathbf{B}$ is only based on *computational*. The euclidean distance between $\mathbf{a}_i$ and $\mathbf{b}_i$ is computed with the formula shown in equation (2.3). After running the same comparison for each of the tokens, the distance between the models $\mathbf{A}$ and $\mathbf{B}$ is then computed as the mean of those tokenwise distances across all the tokens modelled by both: $d(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n} d(\mathbf{a}_i, \mathbf{b}_i)}{N}$. Alternatively, the distances between models could come from procrustes analysis[5], like Wielfaert et al. (2019) do,

[4] Kiela & Clark (2014) also recommend a Correlation similarity.

[5] Run with `vegan::procrustes()` (Oksanen et al. 2020)

which has the advantage of returning a value between 0 and 1. However, this method is much faster and returns comparable results.

$$d(\mathbf{a}_i, \mathbf{b}_i) = \sqrt{\sum_{i=j}^{n} (a_j - b_j)^2} \tag{2.3}$$

2.2.3 Dimensionality reduction for visualization: t-SNE

Dimensionality reduction algorithms try to reduce the number of dimensions of a high-dimensional entity while retaining as much information as possible. In distributional models, they have two main applications: one that reduces thousands of dimensions to a few hundred, and one that reduces them to only two.

The first application of dimensionality reduction, with techniques like SVD (Singular Value Decomposition), is meant to deal with the sparsity of high-dimensional vectors. Due to the frequency distribution discussed above, many words never occur in the vicinity of each other, resulting in many zeros in their context-counting representations and therefore inflated differences between the vectors. In particular, techniques like Latent Semantic Analysis (Landauer & Dumais 1997) are based on the observation that the dimensions obtained from this process are semantically interpretable. It could also be used for token-level spaces, but the comparisons discussed in De Pascale (2019: 246) indicate that they don't necessarily perform better than non reduced spaces. Both dimensionality reduction techniques and neural networks are suggested as ways of condensing very long, sparse vectors (Jurafsky & Martin 2020, Bolognesi 2020). We will not go into the technical aspects because these techniques have not been implemented in the studies described here. Instead, we have compared vectors of different lengths based on other selection methods for the second-order features. Combining them with SVD is a possible avenue for future comparisons.

The second application of dimensionality reduction is used for visualization purposes. A token-by-feature matrix can be understood as a multidimensional space: each of the columns is a dimension of space and the values of cells are the coordinates of the items in each of these dimensions. That is why we can use cosine distances, which measures angles: if we draw a vector from the origin (zero in all dimensions) to the point with those coordinates, it diverges from other vectors with a given angle that grows wider as the vectors diverge, leading to larger cosine distances. We can mentally picture or even draw positions, vectors and angles in up to 3 dimensions, but distributional models have hundreds if not thousands of dimensions. These applications of dimensionality reduction, then, are built to project the distances between items in the multidimensional space to euclidean distances in a low-dimensional space that we can visualize. The different implementations can receive the token-by-feature matrix as input, but will not typically compute cosine distances between the items, so the distance matrix is provided as input instead. The literature tends to go for either multidimensional scaling (MDS) or t-stochastic neighbour embeddings (t-SNE); recently, an interesting alternative called UMAP has been introduced,

which I'll discuss shortly.

The first option, MDS, is an ordination technique, like principal components analysis (PCA). It has been used for decades in multiple areas (e.g. Cox & Cox 2008); its most relevant application for this case, non-metric multidimensional scaling, was developed by Kruskal (1964). It tries out different low-dimensional configurations aiming to maximize the correlation between the pairwise distances in the high-dimensional space and those in the low-dimensional space: items that are close together in one space should stay close together in the other, and items that are far apart in one space should stay far apart in the other. The output from MDS can be evaluated by means of the stress level, i.e. the complement of the correlation coefficient: the smaller the stress, the better the correlation between the measures. Unlike PCA, however, the dimensions are not meaningful *per se*; two different runs of MDS may result in plots that mirror each other while representing the same thing. Nonetheless, the R implementation `vegan::metaMDS()` (Oksanen et al. 2020) rotates the plot so that the horizontal axis represents the maximum variation. In cognitive linguistics literature both metric (Koptjevskaja-Tamm & Sahlgren 2014, Hilpert & Correia Saavedra 2017, Hilpert & Flach 2020) and non-metric MDS (Heylen, Speelman & Geeraerts 2012, Heylen et al. 2015, Perek 2016, De Pascale 2019) have been used.

The second technique, t-SNE (van der Maaten & Hinton 2008, van der Maaten 2014), has also been incorporated in cognitive distributional semantics (Perek 2018, De Pascale 2019). It is also popular in computational linguistics (Smilkov et al. 2016, Jurafsky & Martin 2020); in R, it can be implemented with `Rtsne::Rtsne()` (Krijthe 2018). The algorithm is quite different from MDS: it transforms distances into probability distributions and relies on different functions to approximate them. Moreover, it prioritises preserving local similarity structure rather than the global structure: items that are close together in the high-dimensional space should stay close together in the low-dimensional space, but those that are far apart in the high-dimensional space may be even farther apart in low-dimensional space. Compared to MDS, we obtain nicer, tighter clouds (see Figure 2.2), but the distance between them is less interpretable: even if we trust that tokens that are very close to each other are also similar to each other in the high-dimensional space, we cannot extract meaningful information from the distance *between* these groups.

In addition, it would seem that points that are far away in a high-dimensional space might show up close together in the low-dimensional space (Oskolkov 2021). In contrast, Uniform Manifold Approximation and Projection, or UMAP (McInnes, Healy & Melville 2020), penalizes this sort of discrepancies. It would be an interesting avenue for further research, but a test on the current data did not reveal substantial improvements between t-SNE and UMAP that would warrant the replacement of the technique within the duration of this project (see Figure 2.2 for an example with default parameters. In other models, differences include longer shapes). Other known advantages such as increased speed were not observed in the small samples under consideration — in fact, the R implementation of UMAP (Konopka 2020) was even slower.

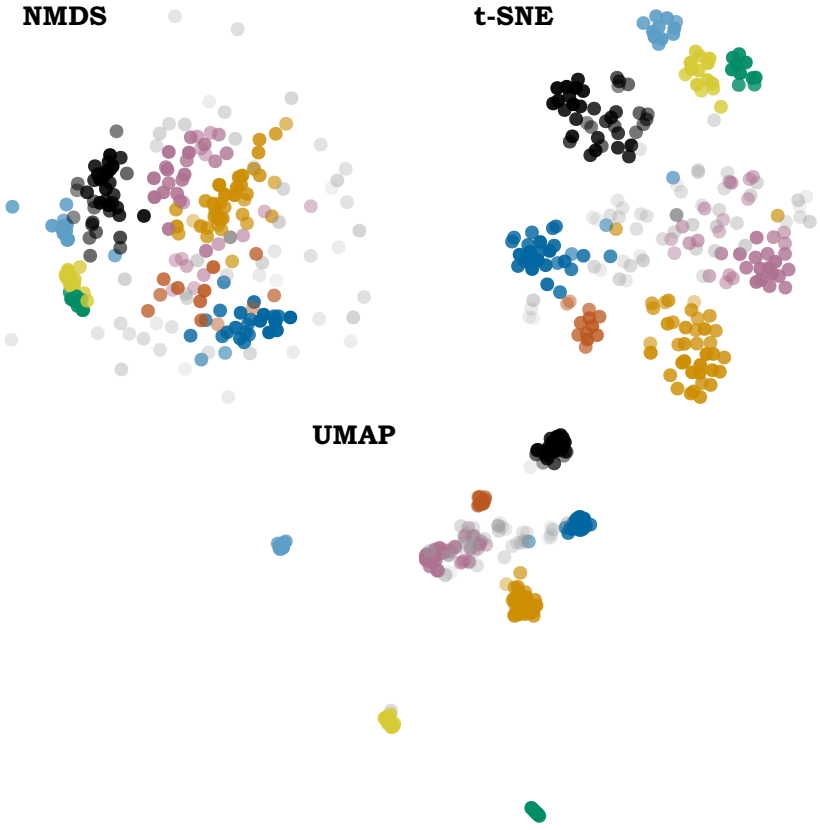

Figure 2.2: Two 2D representations of the same model of *hachelijk* 'dangerous/critical': bound5all-PPMIweight-FOCall. Non-metric MDS on the top left, t-SNE to its right and UMAP at the bottom. Colours indicate HDBSCAN clusters.

Unlike MDS, t-SNE requires setting a parameter called **perplexity**, which roughly indicates how many neighbours the preserved local structure should cover. Low values of perplexity lead to numerous small groups of items, while higher values of perplexity return more uniform, round configurations (Wattenberg, Viégas & Johnson 2016). I have explored perplexity values of 10, 20, 30 and 50, and for this dataset 30 — the default value in the R implementation — has proved to be the most stable and meaningful. Unless otherwise stated, the figures in this text — including Figure 2.1 — will illustrate t-SNE token-level representations with perplexity of 30. To represent distances between models, instead, non-metric MDS is used (only in Section 3.2).

For both MDS and t-SNE we need to state the desired number of dimensions before running the algorithm — for visualization purposes, the most useful choice is 2. Three dimensions are difficult to interpret if projected on a 2D space, such as a screen or paper (Card, Mackinlay & Shneiderman 1999, Wielfaert et

al. 2019). As we mentioned before, the dimensions themselves are meaningless, hence no axes or axis ticks will be included in the plots. However, the scales of both coordinates are kept fixed: given three points $a = (1, 1.5)$, $b = (1, 0.5)$ and $c = (0, 1.5)$, the distance between a and b (1 unit along the x-axis) will be the same as the distance between a and c (1 unit along the y-axis).

2.2.4 Clustering: HDBSCAN

In word sense disambiguation tasks, the vectorial representations of different attestations are clustered into groups of similar tokens. There is a variety of clustering algorithms, appropriate for different kinds of data and structures. I will not offer an overview of the options, but only describe the techniques used in these studies. This section is dedicated to HDBSCAN, the algorithm that returns the coloured clusters in Figures 2.1 and 2.2. Section 2.4 will discuss PAM, which will be uses to select representative models.

Hierarchical Density-Based Spatial Clustering of Applications with Noise, HDBSCAN for the friends (Campello, Moulavi & Sander 2013), is a clustering algorithm, i.e. a procedure to identify groups of similar items that are different from other groups. Unlike its better-known cousins, it does not try to place *all* the items in the sample in different groups, but instead assumes that the dataset might be noisy and that the items may have various degrees of membership to their respective clusters. In addition, as a density-based algorithm, it tries to discriminate between dense areas, i.e. groups of elements that are very similar to each other, from sparse areas, i.e. larger distances between the elements.

In HDBSCAN, the density of the area in which we find a point a is estimated by calculating its core distance $core_k(a)$, which is the distance to its k nearest neighbour, k being a parameter $minPts - 1$. This measure is at the base of the mutual reachability distance, shown in equation (2.4), which is used to compute a new distance matrix for a single-linkage hierarchical clustering algorithm. As a result, the items are organised in a hierarchical tree, from which clusters are selected based on the $minPts$ requirement and their densities. A related notion to $core_k(a)$ is ε, which is defined as the radius around a point in which $minPts - 1$ can be found.

$$d_{mreach}(a, b) = \max(core_k(a), core_k(b), d(a, b)) \tag{2.4}$$

In DBSCAN, we need to set both $minPts$ and a ε threshold; the procedure is different, but its result is equivalent to cutting the hierarchical tree from HDBSCAN at a fixed ε, so that the items above that threshold are discarded as noise, and those below it are grouped into their respective clusters. In contrast, its hierarchical version, HDBSCAN, implements variable thresholds to maximize the stability of the clusters, and therefore only requires us to input $minPts$[6].

In R, the algorithm can be implemented with `dbscan::hdbscan()` (Hahsler & Piekenbrock 2021). Its input can be an item-by-feature matrix or, like in this case, a distance matrix. The output includes, among other things, the cluster assignment, with noise points assigned to a cluster 0, membership probability

[6]For a friendly description of how the algorithm works, I recommend McInnes, Healy & Astels (2016) or even their conference presentations in YouTube.

values, which are core distances normalized per cluster, and ε values, which can be used as an estimate of density.

2.3 Making it your own: parameter settings

Building models implies making a number of choices, from the source of the data and the unit of analysis, to the definition of what counts as context, to the techniques and parameters for visualization and clustering. Making these decisions explicit is crucial: on the one hand, they are necessary to interpret the models themselves, but on the other, they are essential for reproducibility.

For each of the 32 lemmas studied in this project, 200-212 models were created, resulting from the combination of parameter settings meant to define the first-order and second-order contexts. Other choices have been kept fixed across all models in this study, for various reasons, among which are practicality and best performance in the literature. The parameter space is virtually infinite, and exploring even more variations did would have increased the number of models exponentially and made the kind of thorough, qualitative descriptions performed here infeasible. Admittedly, some of the variable parameters could have remained fixed, and some of the fixed parameters could have been varied. Such paths remain open for future projects. In this section, I will discuss these decisions: both the ones that have remained fixed across all the studies and the variations that characterize the multiple models under study. Fixed decisions are not specified in the names of the models; variable parameters, which distinguish models from each other, are coded in their names. When mentioned in further sections, they will be described in three parts: first-order parameters (Section 2.3.2), `PPMI` (Section 2.3.3) and second-order parameters (Section 2.3.4). The values of the parameter settings will be set in `monospace`.

2.3.1 Fixed decisions

First, the analyses presented in this dissertation were performed on a corpus of Dutch and Flemish newspapers: the mode is written and the genre, journalistic. Called the *QLVLNewsCorpus* (De Pascale 2019: 30), it combines parts of the Twente Nieuws Corpus of Netherlandic Dutch (Ordelman et al. 2007) and the yet unpublished Leuven Nieuws Corpus. It comprises articles published between 1999 and 2004, belonging to popular and quality sources for both regions in equal proportion[7] and amounting to a total of 520 million tokens, including punctuation. The corpus was lemmatized and tagged with part-of-speech and dependency relations with Alpino (van Noord 2006).

Second, the unit of analysis, the **lemma**, was defined as a combination of stem and part-of-speech[8]. This applies to items at all levels: the definition of a target, the first-order context features and the second-order features; co-occurrence frequencies and association strength measures are always computed

[7]The newspapers include *Het Laatste Nieuws*, *Het Nieuwsblad*, *De Standaard* and *De Morgen* as Flemish sources and *Algemeen Dagblad*, *Het Parool*, *NRC Handelsblad* and *De Volkskrant* as Netherlandic sources.

[8]The corpus was not lemmatized by stemmed.

with the lemma as unit. Both distributional models and some traditions in collocation research may use word forms instead[9]. On the one hand, stemming and tagging add a layer of processing and interpretation to the text; on the other, word forms of the same lemma tend to behave in different ways. From a lexicographic and lexicological perspective, however, it makes sense to use a lemma as a unit. It is the head of dictionary entries and a more typical unit of linguistic analysis. Furthermore, the (mis)match between word forms and lemmas strongly depends on the language under study: in languages like Spanish, French, Japanese and Dutch, verbs can take many more different forms than in English; conversely, Mandarin lacks morphological variation or even spaces between what could count as words. Concretely, the word form *hoop* in Dutch can correspond to the noun meaning either 'hope' or 'heap', or the verb meaning 'to hope', which can also take other forms such as *hopen*, *hoopt*, *hoopte* and *gehoopt* depending on person, number and tense. Our interest, from a lexicological perspective, lies more in line with studying the behaviour of the noun *hoop* and its meanings, than in conflating the noun with one of verbal forms of the homographic verb.

In that respect, a practical note is in order. The target items under study will be represented with dictionary forms in italics, followed by their approximate English translations in single quotation marks: e.g. *hoop* 'hope/heap', *heilzaam* 'healthy/beneficial', *herstructureren* 'to restructure'. Context words might be represented in figures with the stem and part-of-speech combination used by the lemmas, e.g. *word/verb*, but when mentioned in text the part-of-speech will be excluded, e.g. the passive auxiliary *word*. The English translations will belong to the same part-of-speech as the Dutch term in italics and be as unambiguous as possible. When the Dutch term and its English translations are written in the same way, no translation will be included, e.g. *journalist*.

Third, the context words at both first-order and second-order can, in principle, have any part of speech — except for punctuation — and must have a minimum relative frequency of 1 in 2 million (absolute frequency of 227) after discarding punctuation from the token count in the full *QLVLNewscorpus*. There are 60533 such lemmas in the corpus.

Finally, as described in Section 2.2.1, attraction between types were measured with PPMI, computed on the full co-occurrence matrix, i.e. across the full corpus, based on a symmetric window of 4 tokens to either side, including punctuation; see Turney & Pantel (2010) and Kiela & Clark (2014) for alternatives. Token-level vectors are made by adding the type-level vectors of its context words.[10] For vector comparison, cosine distances were used and then transformed, as explained in Section 2.2.2. The transformed cosine distances were used both as input for visualization techniques and the clustering algorithm. Both non-metric MDS and t-SNE with perplexity values of 10, 20, 30 and 50 were explored, but the analyses discussed in the second part of the dissertation are based on the output from solutions with perplexity 30. Clustering

[9] See Turney & Pantel (2010: 155) and Sahlgren (2008: 47-48) for a discussion and Kiela & Clark (2014: 25) for performance comparisons.

[10] Alternatively, they could be multiplied or averaged, but the results were not all that different.

was performed with HDBSCAN setting $minPts = 8$.

2.3.2 First-order selection parameters

The immediate context of a token is the **first order context**: therefore, first-order parameters are those that influence which elements in the immediate environment of the token will be included in modelling said token. This was performed in two stages: one dependent on whether syntactic information was used, discussed in this section, and one independent of it, shown in Section 2.3.3.

The decisions were based on a mix of literature (e.g. Kiela & Clark 2014), tradition within the Nephological Semantics project, linguistic intuition and generalisations over the annotation of the concordance lines. As we will see in Chapter 4, the manual annotation procedure included selecting words in the context of each token that were the most helpful for the disambiguation. The window spans and dependency information of these chosen context words were used to inform some of the decisions below.

In a first stage, the main distinction is made between models based on bag-of-words (`BOW`), i.e. that do not care about word order or syntactic relationship, and those based on dependency (i.e. syntactic) information. Within the former group, models may vary based on whether sentence boundaries were respected, the length of the window size, and part-of-speech filters. The latter group includes models that select context words based on the distance between them and their target in terms of syntactic relationships (`(LEMMA)PATH` models), and models that find the context word that match specific, predefined templates (`(LEMMA)REL`). Each of these parameters will be described in more detail below.

The first split in `BOW` models distinguishes between those that include words outside the sentence of the target (`nobound`) and those that do not (`bound`). The goal was to make the models more comparable to dependency-based models, which by definition only include words in the same sentence as the target. However, models that only differ with respect to this parameter tend to be extremely similar. More relevant is the window size: models can select context words on a symmetric window of 3, 5, or 10 tokens to either side of the target, including punctuation. Window sizes are typically larger for token-level models than for type-level models (e.g. Schütze 1998, De Pascale 2019), but, at the same time, the great majority of the context words selected in the annotation were within the span of 3 words to either side. In practice, such a small span tends to be too restrictive. Finally, some models refine their first-order selection with part-of-speech filters: `lex` models only include common nouns, adjectives, verbs and adverbs, while `all` models do not implement any restrictions. The selection defined for `lex` was the result of some trial and errors, but could use more refinement for future studies, e.g. expanding the lexical set to proper names, pronouns or only certain prepositions. Moreover, it could be useful to distinguish between modal verbs and auxiliaries, on one side, and other kinds of verbs, information that is not coded in the part-of-speech tags used in this corpus. In practice, `all` models tend to behave similarly to dependency-based models, while `lex` tends to be redundant with PPMI-based selection, which will

be described later. Bag-of-words models will be indicated by a sequence of three values pointing to these three parameters: e.g. `bound5all` indicates a model that respects boundaries, with a window span of 5 words to each side and no part-of-speech filter.

The distinction between `BOW` and dependency-based models doesn't depend so much on which context words are selected but on how tailored the selection is to the specific tokens. For example, a closed-class element like a preposition may be distinctive of particular usage patterns in which a term might occur. However, such a frequent, multifunctional word could easily occur in the immediate raw context of the target without actually being related to it. Unfortunately, just narrowing the window span doesn't solve the problem, since it would also drastically reduce the number of context words available for the token and for any other token in the model. In contrast, we might also be interested in context words that are tightly linked to the target in syntactic terms but separated by many other words in between, but widening the window to include them would imply too much noise for this token and for any other token in the model. A dependency-based model, instead, will only include context words in a certain syntactic relationship to the target, regardless of the number of words in between from a `BOW` perspective. To exemplify, let's look at (4), where *herhalen* 'to repeat', in bold, is the target, and the items in italics where captured by a `PATH` model.

(4) *Als de geschiedenis zich werkelijk mocht* **herhalen**, zijn Vitales dagen *geteld*. (*De Morgen*, 2004-08-02, Art. 98)
'*If* [*the*] *history really* **repeated** *itself*, Vitales' days are *counted*.'

The `PATH` models count the steps between a target and all the words syntactically related to it and base the selection according to that distance. A one-step dependency path is either the head of the target or its direct dependent (the parent or the child, in kinship terms): in the case of (4) this includes the reflexive pronoun *zich* and the modifying adverb *werkelijk* 'really', which depend directly on it *herhalen* 'to repeat', as well as the modal *mocht*, on which the target depends. A two-step dependency path is either the head of the head of the target (grandparent), the dependent of its dependent (grandchild), or its sibling. In (4) this includes the subject *geschiedenis* 'history', because it is linked to the target through the modal, and *Als* 'if'. All `PATH` models include the features in a one-step or two-step path from the target. A three-step dependency path is either the head of the head of the head of the target (great-grandparent), the sibling of the head of its head (great-aunt), the dependent of the dependent of its dependent (great-grandchild), or the dependent of a sibling (niece). In (4) this corresponds to *de* 'the', which depends on *geschiedenis* 'history', and *geteld* 'counted', which *als* 'if' depends on. `PATHselection2` models do not include the three-steps path, and none of the `PATH` models include context words beyond these steps. The threshold was set based on the most frequent syntactic distance between the lemmas from the case studies and the context words selected as relevant for disambiguation. Next to `selection2`, `PATH` models take two more formats. While `selection3` models include context words up to 3 steps away from the target, `PATHweight` models also incorporate

the distance information and give more weight to context words that are more directly closely to the target in the syntactic path.

Finally, `REL` models base their selection on specific, predefined patterns. For these purpose, templates tailored to the parts of speech of the target were designed, based on the relationships between the annotated types and the context words selected as most informative during the annotation process. The most restrictive model, `RELgroup1`, selects the following patterns:

- For nouns: modifiers and determiners of the target; items of which the target is modifier or determiner, and verbs of which the target is object or subject.
- For adjectives: nouns modified by the target and direct modifiers of it (except for prepositions); subject and direct objects of the verbs of which the target is direct modifier or predicate complement, with up to one modal or auxiliary in between.
- For verbs: direct objects; active and passive subjects (with up to two modals for the active one); reflexive complement, and prepositions depending directly on the target.

It is typically too restrictive: for many lemmas, it is responsible for the loss of a large proportion of tokens which do not have context words that match these patterns, while the remaining tokens often have only one or two context words left. The `RELgroup2` models expand the selection as follows:

- For nouns: conjuncts of the target (with or without conjunction); objects of the modifier of the target, and items on which the target depends via a modifier.
- For adjectives: object of the preposition modifying the target; conjunct of the target (with or without conjunction); prepositional object of verb modified by target (as modifier or prepositional complement).
- For verbs: conjuncts of the target; complementizers; nouns depending through a preposition; verbal complements, and elements of which the target is a verbal complement.

Finally, nouns also have a `RELgroup3` setting that incorporates the following relations:

- Objects and modifiers of items of which the target is subject or modifier; subjects and modifiers of items of which the target is object or modifier; modifiers of the modifiers of the target, and items of whose modifier the target is modifier.

All the first-order parameters procure filters to select the context words in the environment of each token that will be used to model it. Alternatively, dependency information could have been included as a feature or dimension. For example, instead of selecting *zich* 'itself' as context word of the token in (4) based on its bag-of-word distance, part-of-speech filter or dependency relation to the target, we could use `(zich, se)` i.e. "has *zich* as reflexive subject" as a first-order feature. Its type-level vector then would have information on all the

other verbs that take *geschiedenis* 'history' as its subject. For technical and practical reasons, this was not implemented in the studies discussed here, but would be a fruitful path for further research.

In the remainder of this dissertation, `BOW` will be used to refer to all bag-of-words based models, as opposed to the dependency-based models; `PATH` and `REL` will also be umbrella terms for the models that use the different kinds of dependency-based selection, and more specific terms, e.g. `PATHweight` will be used for finer grained distinctions.

2.3.3 PPMI selection and weighting

The `PPMI` parameter[11] is taken outside the set of first-order parameters because it applies to both `BOW` and dependency-based models, although it also affects the selection of first order context words. The rationale behind it is that words in the vicinity of the target token, regardless of their part-of-speech and distance, are not equally informative of the meaning of the target. For example, in (4) *geschiedenis* 'history' and *zich* 'itself' are more informative of the meaning of *herhalen* 'to repeat' than *werkelijk* 'really' or *als* 'if'. Association strength measures like PPMI could then be used to give more influence to the more informative context words; indeed, given a symmetric windowsize of 4 for the PPMI computation in the *QLVLNewsCorpus*, the PPMI of *geschiedenis* 'history' and *zich* 'itself' with *herhalen* 'to repeat' are 3.79 and 1.97 respectively, while the values for *werkelijk* 'really' and *als* 'if' are 0.06 and 0.112. Heylen et al. (2015) weight the contribution of each context word by their PPMI with the target, and De Pascale (2019) adds PPMI and LLR (log-likelihood ratio) thresholds to the selection of context words. However, these measures are meant to represent the relationship *between* types, not to distinguish between senses of the same type: a context word may be indicative of a sense of a word and yet not be particularly attracted to the word as a whole. An example is the English verb *to go*, which due to its high frequency does not have a strong attraction to the noun *church*, and yet is necessary to distinguish the specific sense of 'religious service' in *to go to church*.

For that reason, models can take three different settings in relation to the `PPMI` parameter: `weight`, `selection` and `no`. Both `weight` and `selection` apply an additional filter to the output from the first-order parameters and only select the context words with a positive PMI with the target. They are distinct from the `PPMIno` models, which do not apply such thresholds. The difference between the first two is that `weight` also multiplies the type-level vector of each context word by their PMI with the target, giving words that are more strongly associated to the target type a greater impact in the final vector of the target. The three settings are applied to each of the models resulting from the first-order combinations, with one exception: `PATHweight` models do not combine with `PPMIweight`.

[11] I use verbatim to refer to this parameter for two reasons. First, because, like `PATH`, it is also a value: the parameter itself is the association-strength-based filtering or weighting, but it was fixed to PPMI. Second, this way it is easier to distinguish the parameter setting from PPMI as a measure. It is not the best idea, but so it is coded into the current names of the models.

2.3.4 Second-order selection

The selection of second-order features influences the shape of the vectors: how the selected first-order features are represented. Next to the fixed window size and association measure used to calculate the values of the vectors, there are two variable parameters. First, a part-of-speech filter may be applied. When its value is `nav`, second-order features are extracted from a pool of 13771 nouns, adjectives and verbs used in De Pascale (2019)[12]. The alternative, `all`, applies no further filters. Second, we might reduce the length of the vector, i.e. the number of second-order features. One of the values, `5000`, selects the 5000 most frequent features from the pool remaining after the part-of-speech filter. Pilot studies have also explored models with 10000 dimensions, but they are very similar to the ones with 5000 dimensions.[13] The other value for the vector length is `FOC`, which stands for "first-order context", and it uses the union of first-order context words for all tokens as second-order dimensions. As a consequence, the second-order dimensions are tailored to the context of the sample, not necessarily so frequent, and their numbers remain in the hundreds, rarely surpassing 1500. In practice, there is not much of a difference between models with different second-order parameters, except for `5000all` models, which tend to perform the worst. Examination of the distance matrices between the type-level vectors of the context words reveals that the cosine distances between all of them are really large, probably due to the sparseness of the vectors. I that sense, it would be interesting to compare SVD matrices based on the `5000` models with the already smaller (and presumably denser) `FOC` models.

2.4 The chosen ones: PAM

The multiple variable parameters return a large number of models: 212 for each of the nouns — because of the additional `REL` templates — and 200 for verbs and adjectives. As we will see in Chapter 3, we can combine distances between the models with dimensionality reduction techniques to represent the similarities between the models on a 2D space. In addition, if we only wanted to evaluate the models in relation to the manual annotation, we could rank the accuracy of their clustering solutions. However, if we want to understand the qualitative effect of the parameter settings on the modelling, and especially if we do not consider the manual annotation as a ground truth, we need to examine clouds individually, and it is not feasible for a human to look at each of the hundreds of models of each lemma.

One approach for an efficient exploration of the parameter space is to identify the settings that make the greater differences between models. For example, if we see that models with different `PPMI` settings are more different from each other than models with different vector-length settings, we would prioritize looking at models that differ on the former parameter, setting the latter to a constant value. Unfortunately, the quantitative effect of parameters is not

[12]The selection was originally made to ensure an unbiased regional distribution of the vectors for the lectometric studies performed in De Pascale (2019).

[13]Kiela & Clark (2014) discourage using vectors with more than 50,000 dimensions.

so straightforward. First, the parameters that tend to make a big difference in the modelling include the choice between dependency and `BOW` and, within it, both window size and part-of-speech filters, as well as the distinction between `REL` and `PATH`. The resulting combinations are still too numerous to examine simultaneously (see Chapter 3). Second, the relevant parameters interact with each other: `PPMI` often makes little difference among `lex` models — it tends to be redundant, since the open-class items captured by `lex` tend to have higher PPMI — but it makes a greater difference among `all` or dependency-based models. Finally, the various parameter settings do not have the same impact within each lemma, so they have to be revised for each of the lemmas under study.

The approach based on the quantitative effect of parameter settings on the distances between models does reduce the number of models to examine, but not to a great degree. Given the limited number of models that we can look at simultaneously while still making sense of them — around 8 or 9 — and the need to cover multiple combinations of these strong parameters, we would still need to look at four or five partially overlapping sets of 8-9 models per lemma. For example, a set of 9 models could be generated by taking `bound3` and `bound10` models with `PPMIweight` and `FOCnav` second-order vectors, in order to look at the effect of part-of-speech filter with little window-size variation, `PATH` and `REL`. Then, `PPMIweight` could be switched for `PPMIno` to look at the effect in the new conditions, resulting in 9 other models. If the effect is indeed different, which is likely, a different set of 8-9 models could then be generated with different values of `PPMI`, while keeping the part-of-speech to a constant value. These groups are not maximally different from each other: due to the interaction between parameters, many models are extremely similar, and a proper qualitative description becomes challenging. Moreover, a given set of models could reveal a pattern that was not captured in a previous set of models, and the researcher might want to go back and look for it.

An alternative approach is to use a clustering algorithm that, next to selecting groups of similar models, identifies the models that represent each of the clusters. **Partition Around Medoids**, or PAM (Kaufman & Rousseeuw 1990), implemented in R with `cluster::pam()` (Maechler et al. 2021), does exactly that. Unlike HDBSCAN and other clustering algorithms, it requires us to set a number of clusters beforehand, and then tries to find the organization that maximizes internal similarity within the cluster and distances with other clusters. For our purpose, we have settled for 8 medoids for each lemma. The number is not meant to achieve the best clustering solutions — no number could be applied to all the lemmas with equal success, given their variability in the differences between the models. The goal, instead, is to have a set of models that is small enough to visualize simultaneously (on a screen, in reasonable size) and big enough to cover the variation across models. For some lemmas, there might not be that much variation, and the **medoids**, i.e. the representative models, might be redundant with each other. However, as long as we can cover (most of) the visible variation across models and the medoids are reasonably good representatives of the models in their corresponding clusters, the method is fulfilling its goal.

The representativeness of medoids for the lemmas studied here has been tested in different ways. We don't require the clusters of models to be different from each other, as long as the medoids represent them properly. Instead, the priority was to check for patterns within the models represented by each medoid, e.g. in terms of accuracy towards annotated senses. For example, if a medoid tends to group senses together very well (measured for example with `kNN` and `SIL`, as explained in Chapter 5 applied to clustering solutions), the models it represents have similar tendencies as well. More importantly, different patterns previously identified in the plots while exploring the models with the first approach were looked for in the medoid selection, to corroborate that the medoids covered at least as much variation as the more time- and energy-consuming approach. All such patterns were found. In addition, small random samples within each cluster of models were visually scanned — but not thoroughly examined — to assess their similarity to their representative medoid. In the great majority of the cases the comparison was satisfactory. This has a wonderful effect on the visual exploration, because it lets us focus on 8-9 models that are quite different from each other instead of multiple sets of models with less variation. Visually, the medoids approach is more informative and less tiresome.

As a result from these explorations, the qualitative analyses will be based on medoids: representative models selected by PAM. While this is a clustering algorithm, in order to avoid confusion with clusters of tokens, which take centre stage, I will avoid referring to the clusters of models as such — or, if I do, I will specify that they are clusters *of models*. The preferred name will be "the models represented by the medoid". Given that the only clustering algorithm used on the tokens is HDBSCAN, *medoid* will always refer to a representative model.

2.5 Summary

The process through which token-level vector space models and the clouds studied here in particular are created, takes a number of transformative steps. In this chapter we have broken down this process and detailed the layers of mathematical and linguistic processing lying between the raw corpus and the final clouds. Next to an overall description of the workflow, the technical background of the most important aspects was introduced in some detail. Afterwards, I explained the parameter settings that characterize the models analyzed in this project. Choices have been made and alternatives have been suggested: the path taken here was one out of so many possible alternatives. In fact, at the core of this research project is the exploration of alternatives, the investigation of the effect of the variable parameters on the final linguistic representation, and the search for clues, guidelines, a recipe for the clouds we seek. This exploration combines quantitative techniques — the heart of the process of cloud creation — with qualitative analyses meant to describe what and how the clouds are really modelling.

By combining the vector representations with visualization techniques and/or clustering algorithms, we can make sense of patterns that would

otherwise escape us. Visual analytics provides us with tools to explore the output in comfortable, intuitive — but sometimes deceiving — ways. In the next chapter, we will look at the two visualization tools developed within the larger project of Nephological Semantics to enable and support these qualitative analyses.

Chapter 3

Visualization tools

Clouds are the prime matter of these study. They are condensed, information-rich representations of patterns found in a corpus and should, according to the Distributional Hypothesis, tell us something about the meaning of the words under examination. But they don't tell us anything by themselves: we need to develop and implement tools to extract this information. Chief among these tools is a web-based visualization tool (Montes & QLVL 2021), originally developed by Thomas Wielfaert within the Nephological Semantics project (see Wielfaert et al. 2019), and then continued by myself[1]. In this chapter we will present its rationale and the features it offers, as an elaboration of Montes & Heylen (Submitted).

Section 3.1 will offer an overview of the rationale behind the tool and the minimal path that a researcher could take through its levels. Sections 3.2 through 3.4 will zoom in on each of the levels, describing the current features and those that are still waiting on our wish list. Section 3.5 follows with the description of a ShinyApp (Chang et al. 2021): an extension[2] to the third level of the visualization with additional features tailored to exploring the relationship between the 2D representations and the HDBSCAN output. Finally, we conclude with a summary in Section 3.6.

3.1 Flying through the clouds

The visualization tool described here, which I will call *NephoVis*, was written in Javascript, making heavy use of the D3.js library, which was designed for beautiful web-based data-driven visualization (Bostock, Ogievetsky & Heer 2011). The D3 library allows the designer to link elements on the page, such as circles in an SVG, dropdown buttons and titles, to data structures such as arrays and data frames, and manipulate the visual elements based on the values

[1] The GitHub repository is linked to Zenodo, so that the released versions can be stored and identified with a DOI. Unfortunately, even though the foundations of the code were set by Thomas Wielfaert, because of how the current repository came to be, he has no history as contributor and therefore is not assigned as author in the tool's citation.

[2] Currently available at https://marianamontes.shinyapps.io/Level3/

of the linked data items. In addition, it offers handy functions for scaling and mapping, i.e. to fit the relatively arbitrary ranges of the coordinates to pixels on a screen, or to map a colour palette[3] to a set of categorical values.

As we have seen in Chapter 2, the final output of the modelling procedure is a 2D representation of distances between tokens, which can be visualized as a scatterplot. Crucially, we are not only interested in exploring individual models, but in comparing a range of models generated by variable parameters. Section 2.2.2 discussed a procedure to measure the distance between models, which can be provided as input for non-metric MDS, and Section 2.4 presented the technique used to select representative models, or medoids. As a result, we have access to the following datasets for each of the lemmas:

- A distance matrix between models.
- A data frame with one row per model, the NMDS coordinates based on the distance matrix, and columns coding the different variable parameters or other pieces of useful information, such as the number of modelled tokens.
- A data frame with one row per token, 2D coordinates for each of their models and other information such as sense annotation (see Chapter 4), country, type of newspaper, selection of context words and concordance line.
- A data frame with one row per first-order context word and useful frequency information.

In practice, the data frame for the tokens is split in multiple data frames with coordinates corresponding to different dimensionality reduction algorithms, such as NMDS and t-SNE with different perplexity values, and another data frame for the rest of the information. In addition, one of the most recent features of the visualization tool includes the possibility to compare an individual token-level model with the representation of the type-level modelling of its first-order context words. However, this feature is still under development within NephoVis and can be better explored in the ShinyApp extension (Section 3.5).

In order to facilitate the exploration of all this information, NephoVis is organized in three levels, following Shneiderman's Visual Information Seeking Mantra: "Overview first, zoom and filter, then details-on-demand" (1996: 97). The core of the tool is the interactive, zoomable scatterplot, but its goal and functionality is adapted to each of the three levels. In Level 1 the scatterplot represents the full set of models and allows the user to explore the quantitative effect of different parameter settings and to select a small number of models for detailed exploration in Level 2. This second level shows multiple token-level scatterplots — one for each of the selected models —, and therefore offers the possibility to compare the shape and organization of the groups of tokens across different models. By selecting one of these models, the user can examine it in Level 3, which focuses on only one at a time. Shneiderman (1996)'s mantra

[3]While D3 offers a variety of useful colour palettes, the visualization currently relies on a — slightly adapted — colorblind-friendly scale by Okabe & Ito (2002). The default colour palette for most of the figures in this disseration make use of the same palette, via the R package `colorblindr` (McWhite & Wilke 2020).

underlies both the flow across levels and the features within them: each level is a zoomed in, filtered version of the level before it; the individual plots in Levels 1 and 3 are literally zoomable; and in all cases it is possible to select items for more detailed inspection. Finally, a number of features — tooltips and pop-up tables — show details on demand, such as the names of the models in Level 1 and the context of the tokens in the other two levels.

Figure 3.1: Portal of https://qlvl.github.io/NephoVis/ as of July 2021.

Currently, https://qlvl.github.io/NephoVis/ hosts the portal shown in Figure 3.1, which eventually leads the user to the Level 1 page for the lemma of their choice[4], shown in Figure 3.2 and described in more detail in Section 3.2. By exploring the scatterplot of models, the user can look for structure in the distribution of the parameters on the plot. For example, colour coding may reveal that models with nouns, adjectives, verbs and adverbs as first-order context words (`lex`) are very different from those without strong filters for part-of-speech, because mapping these values to colours reveals distinct groups in the plot. In contrast, mapping the sentence boundaries restriction (`bound`/`nobound`) might result in a mix of dots of different colours, like a fallen bag of M&M's, meaning that the parameter makes little difference. Depending on whether the user wants to compare models similar or different to each other, or which parameters they would like to keep fixed, they will use individual selection or the buttons to choose models for Level 2. The **Select medoids** button[5] quickly identifies the predefined medoids. By clicking on the

[4]By knowing the lemma, it is possible to go directly to the Level 1 page by replacing *lemma* in https://qlvl.github.io/NephoVis/level1.html?type=lemma with the corresponding name of the lemma, e.g. *heffen*.

[5]Incorporating this feature is less scalable than the dropdown menus or even the checkbox buttons; it works with the current pipeline, but is not so straightforward to adapt to new data that does not follow the exact same pipeline. Flexibilizing the features to allow for missing

LEVEL 2 button, Level 2 is opened in a new tab, as shown in Figure 3.3.

In Level 2, the user can already compare the shapes that the models take in their respective plots, the distribution of categories like sense labels, and the number of lost tokens. If multiple dimensionality reduction techniques have been used, the **Switch solution** button allows the user to select one and watch the models readjust to the new coordinates in a short animation. In addition, the **Distance matrix** button offers a heatmap of the pairwise distances between the selected models. Section 3.3 will explore the most relevant features that aid the comparison across models, such as brushing sections of a model to find the same tokens in different models and accessing a table with frequency information of the context words co-occurring with the selected tokens. Either by clicking on the name of a model or through the **Go to model** dropdown menu, the user can access Level 3 and explore the scatterplot of an individual model. As Section 3.4 will show, Level 2 and Level 3, both built around token-level scatterplots, share a large number of functionalities. The difference lies in the possibility of examining features particular of a model, such as reading annotated concordance lines highlighting the information captured by the model or selecting tokens based on the words that co-occur with it. In practice, the user would switch back and forth between Level 2 and Level 3: between comparing a number of models and digging into particular models.

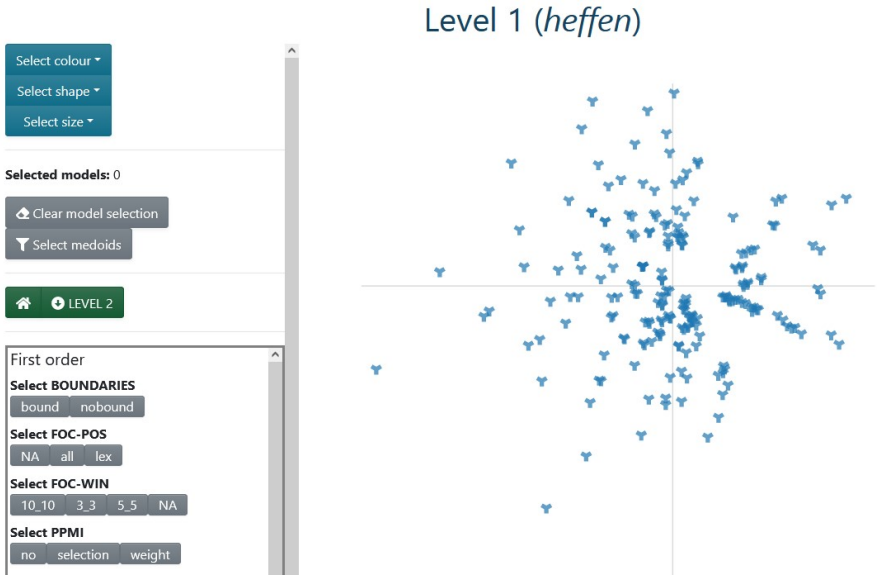

Figure 3.2: Level 1 for *heffen* 'to levy/to lift'.

Before going into the detailed description of each level, a note is in order. As already mentioned in Section 2.2.3, the dimensions resulting from NMDS —

data frames is one of the items on the wish list. Ideally, future versions will implement algorithms such as PAM to compute on the fly as well.

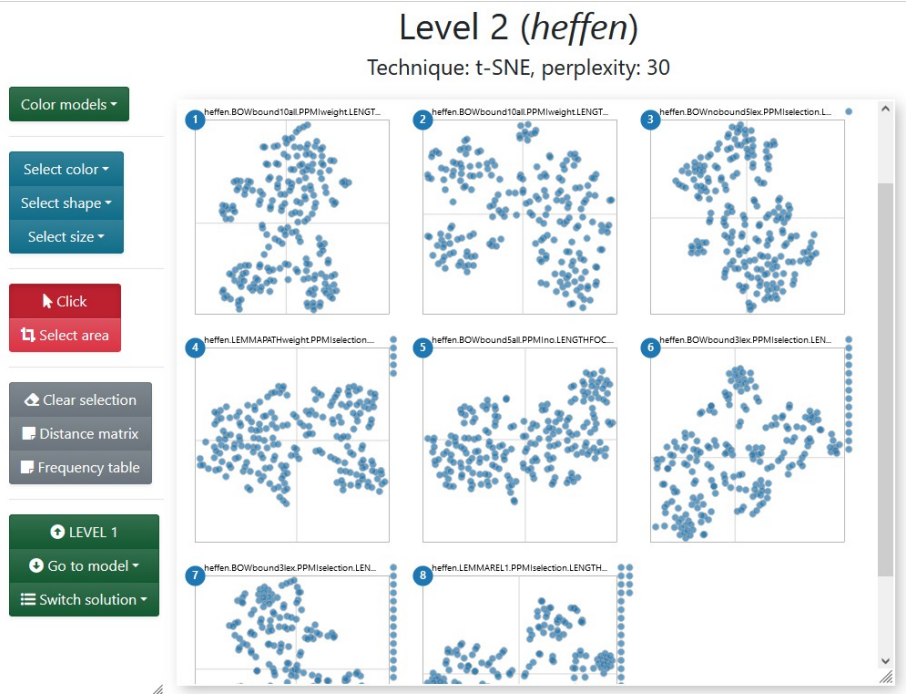

Figure 3.3: Level 2 for the medoids of *heffen* 'to levy/to lift'.

used in all levels — and t-SNE — used in levels 2 and 3 — are not meaningful. In consequence, there are no axes or axis ticks in the plots. However, the units are kept constant within each plot: one unit on the x-axis has the same length in pixels as one unit on a y-axis within the same scatterplot; this equality, however, is not valid across plots. Finally, the code is open-source and available at https://github.com/qlvl/NephoVis.

3.2 Level 1

The protagonist of Level 1 is an interactive zoomable scatterplot where each glyph, by default a steel blue wye ("Y"), represents one model. This scatterplot aims to represent the similarity between models as coded by the NMDS output and allows the user to select the models to inspect according to different criteria. Categorical variables (e.g. whether sentence boundaries are used) can be mapped to colours and shapes, as shown in Figure 3.4, and numerical variables (e.g. number of tokens in the model) can be mapped to size.

A selection of buttons on the left panel, as well as the legends for colour and shape, can be used to filter models with a certain parameter setting. These options are generated automatically by reading the columns in the data frame of models and interpreting column names starting with `foc_` as representing

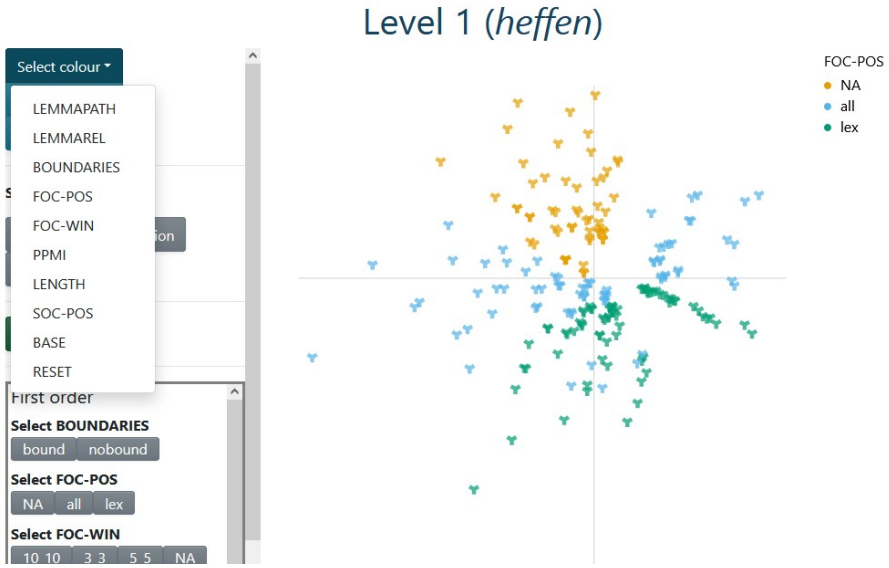

Figure 3.4: Level 1 for *heffen* 'to levy/to lift'; the plot is colour-coded with first-order part-of-speech settings; `NA` stands for the dependency-based models.

first-order parameter settings, and those starting with `soc_` as second-order parameter settings. Different settings of the same parameter interact with an `OR` relationship, since they are mutually exclusive, while settings of different parameters combine in an `AND` relationship. For example, by clicking on the grey `bound` and `lex` buttons on the bottom left, only `BOW` models with part-of-speech filter and sentence boundary limits[6] will be selected. By clicking on both `lex` and `all`, all `BOW` models are selected, regardless of the part-of-speech filter, but dependency-based models (for which part-of-speech is not relevant) are excluded. A counter above, circled in Figure 3.5, keeps track of the number of selected items, since Level 2 only allows up to 9 models for comparison[7]. This procedure is meant to aid a selection based on relevant parameters, as described in Section 2.4. In Figure 3.5, instead, the **Select medoids** button was used to quickly capture the medoids obtained from PAM. Models can also be manually selected by clicking on the glyphs that represent them.

[6]Notice that `bound` itself, while a `BOW` parameter value, also includes the dependency-based models, since they are automatically limited to sentence boundaries.

[7]The original design, found in http://tokenclouds.github.io/LeTok/, allowed for larger selections; only 9 models would be actually shown in Level 2, but it would also be possible to remove some of them and make place to the models left on the waiting list. This makes sense when models are selected individually and in a particular order, i.e. by clicking on them, but not so much for selections based on other criteria that we want to explore simultaneously.

Figure 3.5: Level 1 for *heffen* 'to levy/to lift' with medoids highlighted.

3.3 Level 2

Level 2 shows an array of small scatterplots, each of which represents a token-level model. The glyphs, by default steel blue circles, stand for individual tokens, i.e. attestations of the chosen lemma in a given sample. The original code for this level was inspired by Mike Bostock's brushable scatterplot matrix, but it is not a scatterplot matrix itself, and its current implementation is somewhat different.

The dropdown menus on the sidebar (Figure 3.3) read the columns in the data frame of variables, which can include any sort of information for each of the tokens, such as sense annotation, sources, number of context words in a model, concordance lines, etc. Categorical variables can be used for colour- and shape-coding, as shown in Figure 3.8, where the senses of the chosen lemma are mapped to colours; numerical variables, such as the number of context words selected by a given lemma, can be mapped to size. Note that the mapping will be applied equally to all the selected models: the current code does not allow for variables — other than the coordinates themselves — to adapt to the specific model in each scatterplot. That is the purview of Level 3.

Before further examining the scatterplots, a small note should be made about the distance matrix mentioned above. The heatmap corresponding to the medoids of *heffen* 'to levy/to lift' is shown in Figure 3.6. The NMDS representation in Level 1 tried to find patterns and keep the relative distances between the models as faithful to their original positions as possible, but such a transformation always loses information. Given a restricted selection of models, however, the actual distances can be examined and compared more easily. A heatmap maps the range of values to the intensity of the colours, making patterns of similar/different objects easier to identify. For example, Figure 3.6 shows that the sixth medoid is very different from all the other medoids except from the seventh, and that the second medoids is quite different from all the others except the first. Especially when the model selection followed a criterion based on strong parameter settings, e.g. keeping `PPMI` constant to look at the interaction between window size and part-of-speech filters, such a heatmap could reveal patterns that are slightly distorted by the dimensionality reduction in Level 1 and even hard to pinpoint from visually comparing the scatterplots.

But even with the medoid selection, which aims to find representatives that are maximally different from each other (or at least that are the core elements of maximally different *groups*), the heatmap can show whether some medoids are drastically *more* different, or conversely, similar to others. As a reference, the heatmap is particularly useful to check hypotheses about the visual similarity of models. For example, unlike with *heffen* 'to levy/to lift' in Figure 3.8, if we colour-code the medoids of *haten* 'to hate' with the manual annotation (Figure 3.7), all the models look equally messy. As we will see below, we can brush over sections of the plot to see if, at least, the tokens that are close together in one medoid are also close together in another (spoiler alert: not the case). The heatmap of distances confirms that not all models are equally different from each other, but indeed, each of them are messy in their own particular way.

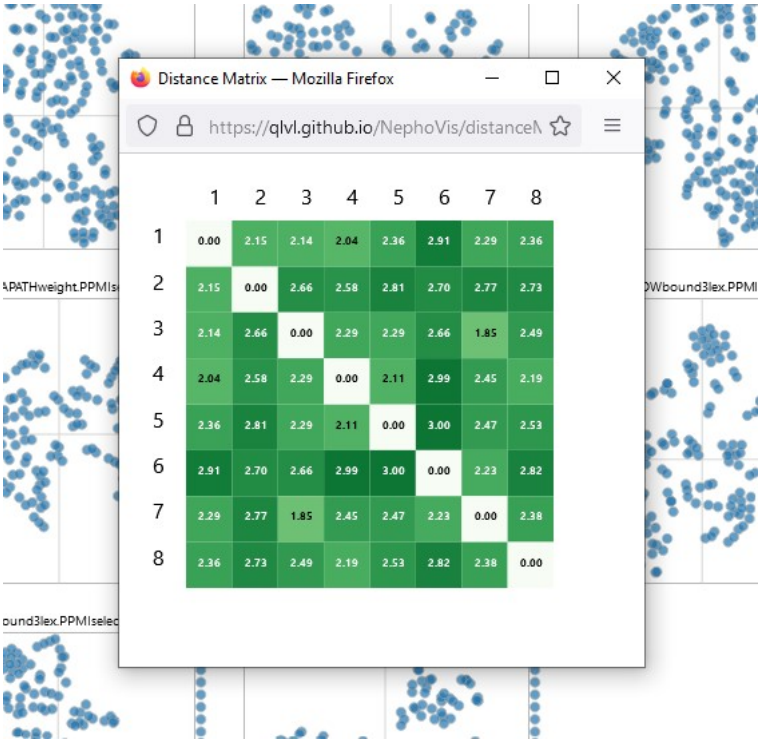

Figure 3.6: Heatmap of distances between medoids of *heffen* 'to levy/to lift'.

Next to the colour-coding, Figure 3.8 also illustrates how hovering over a token shows the corresponding identifier[8] and concordance line. Figure 3.9, on

[8]The identifier of a token includes four main pieces of information separated by slashes. The first two, stem and part-of-speech (*hef* and *verb* in the example) indicate the target lemma. The third section points to the filename from which the token was extracted. The filenames from this corpus have at least three sections split by underscores: the name of the newspaper (*De Volkskrant*), the date of publication in YYYY-MM-DD format (2001-06-21) and the number of the article, among those harvested for the corpus (36). The final part points to the index of the token in the article including punctuation: in this case, the word form *hieven*

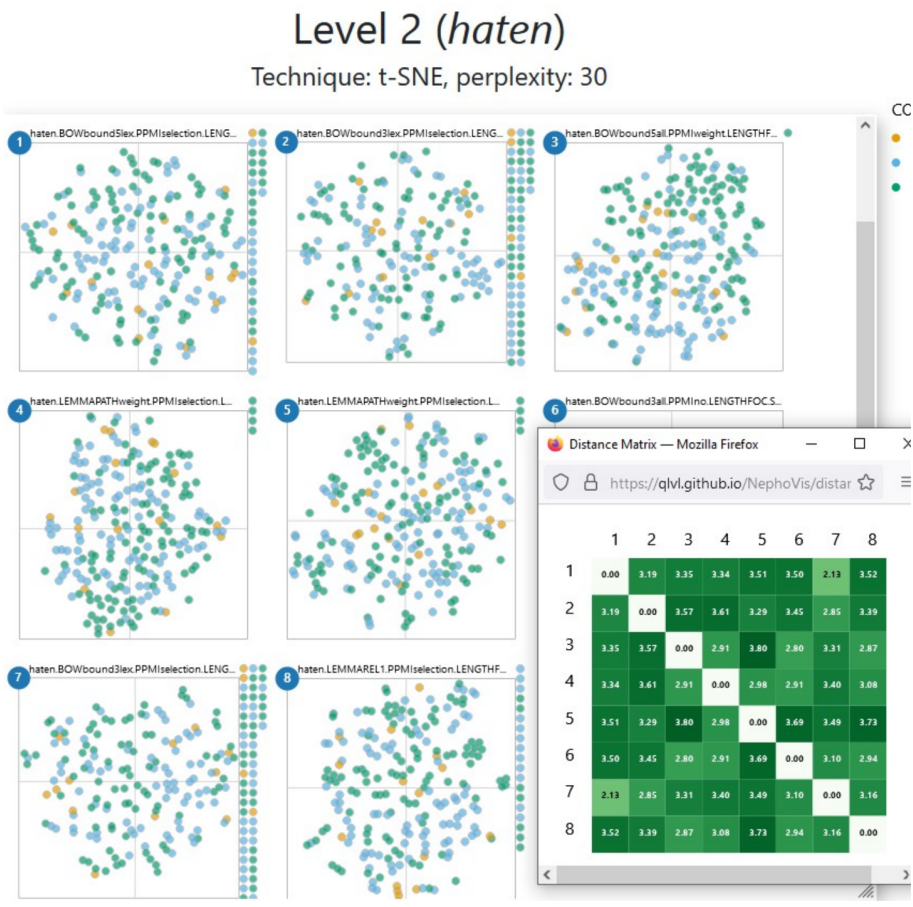

Figure 3.7: 2D representation of medoids of *haten* 'to hate', colour-coded with senses, next to the heatmap of distances between models.

the other hand, showcases the brush-and-link functionality. By brushing over a specific model, the tokens found in that area are highlighted and the rest are made more transparent. Such a functionality is missing from Level 1, but is also available in Level 3. Level 2 enhances the power of this feature by selecting the same tokens in the rest of the models, whatever area they occupy. Thus, we can see whether tokens that are close together in one model are still close together in a different model, which is specially handy in more uniform plots, like the one for *haten* 'to hate' in Figure 3.7. Figure 3.9 reveals that the tokens selected in the second medoid are, indeed, quite well grouped in the other five medoids around it, with different degrees of compactness. It also highlights two glyphs on the right margin of the bottom right plot. In Level 2, this margin

(third person plural preteritum of *heffen*, '(they) lifted') around which the concordance line is built is the 163rd token in its file.

gathers all the tokens that were selected for modelling but were lost by the model in question due to lack of context words. In this case medoid 6, with a combination of `bound3lex` and `PPMIselection`, is extremely selective, and for a few tokens no context words could be captured.

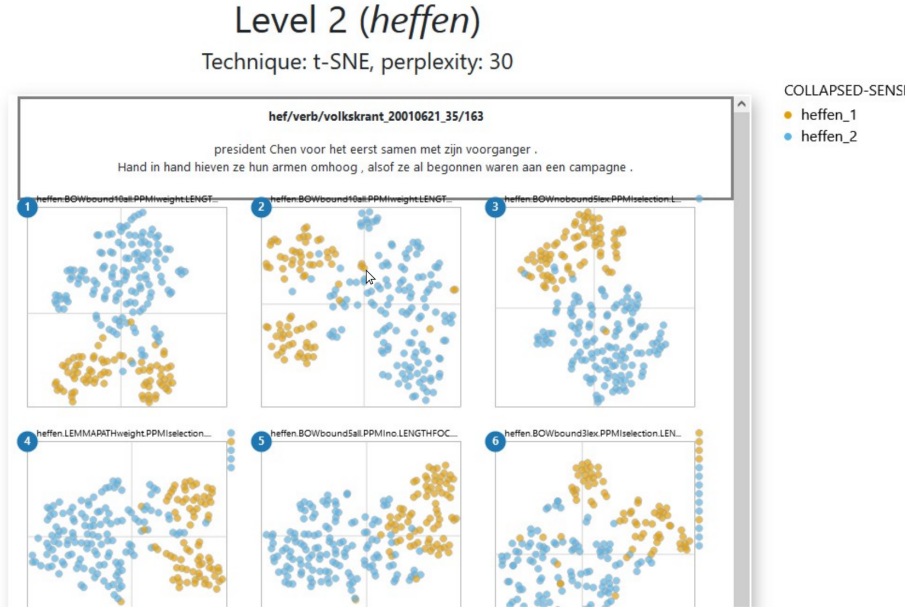

Figure 3.8: Level 2 for the medoids of *heffen* 'to levy/to lift', colour-coded with categories from manual annotation. Hovering over a token shows its concordance line.

In any given model, we expect tokens to be close together because they share a context word, and/or because their context words are distributionally similar to each other: their type-level vectors are near neighbours. Therefore, when inspecting a model, we might want to know which context word(s) pull certain tokens together, or why tokens that we expect to be together are far apart instead. For individual models, this can be best achieved via the ShinyApp described in Section 3.5, but NephoVis also includes features to explore the effect of context words, such as frequency tables. In Level 2, while comparing different models, the frequency table has one row per context word and one or two columns per selected model, e.g. the medoids. Such a table is shown in Figure 3.10. The columns in this table are all computed by NephoVis itself based on lists of context words per token per model. Next to the column with the name of the context word, the default table shows two columns called "total" and two per model, headed by the corresponding number and either a "+" or a "-" sign. The "+" columns indicate how many *of the selected tokens* in that model co-occur with the word in the row; the "-" columns indicate the number of non selected tokens that co-occur with the word. The "total"

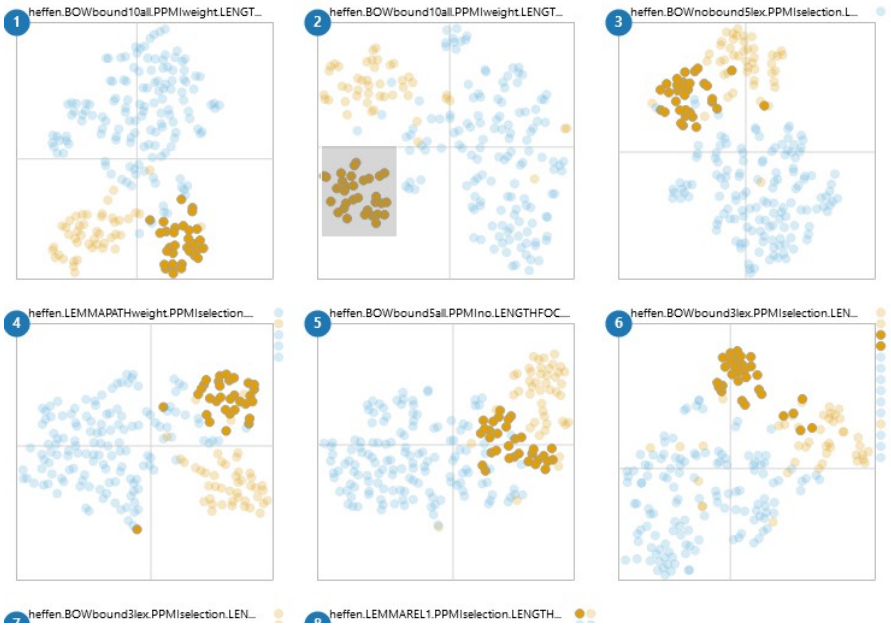

Figure 3.9: Level 2 for the medoids of *heffen*, colour coded with categories from manual annotation. Brushing over an area in a plot selects the tokens in that area and their positions in other models.

columns indicate, respectively, the number of selected or non-selected tokens for which that context word was captured by at least one model. Here it is crucial to understand that, when it comes to distributional modelling, a **context word** is not simply a word that can be found in the concordance line of the token, but an item captured by a given model. Therefore, a word can be a context word in a model, but be excluded by a different model with stricter filters. For example, the screenshot[9] in Figure 3.10 gives us a glimpse of the frequency table corresponding to the tokens selected already in Figure 3.9. The most frequent context word for the 31 selected tokens, i.e. the first row of the table, is the noun *glas* 'glass', which is used in expressions such as *een glas heffen op iemand* 'to toast for someone, lit. to lift a glass on someone'. The columns for models 1 an 2 show that *glas* 'glass' was captured by those models for all 31 selected tokens. In column 3, however, the positive column reads 29, which indicates that the model missed the co-occurrence of *glas* 'glass' in two of the tokens. The names on top of the plots reveal that the first two models have a window size of 10, while the third restricts it to 5, meaning that in the two missed tokens *glas* 'glass' occurs 6 to 10 slots away from the target. These are

[9]The full picture is impractical to include in a printed text; it is recommended to explore the tool interactively instead.

most likely the orange tokens a bit far to the right of the main highlighted area in the third plot. Finally, in the fourth model, which is hidden behind the table, *glas* 'glass' is missed from one of the 31 tokens but captured in 2 tokens that were excluded from the selection. If we moved the window of the table we would see that this is a `PATHweight` model: the missed co-occurrence must be within the BOW window span but too far in the dependency path, wile the two captured co-occurrences in the "-" column must be within three steps of the dependency path but beyond the BOW window span of 10. This useful frequency information is available for all the context words that are captured by at least one model in any of the selected tokens. In addition, the **Select information** dropdown menu gives access to a range of transformations based on these frequencies, such as odds ratio, Fisher Exact and cue validity.

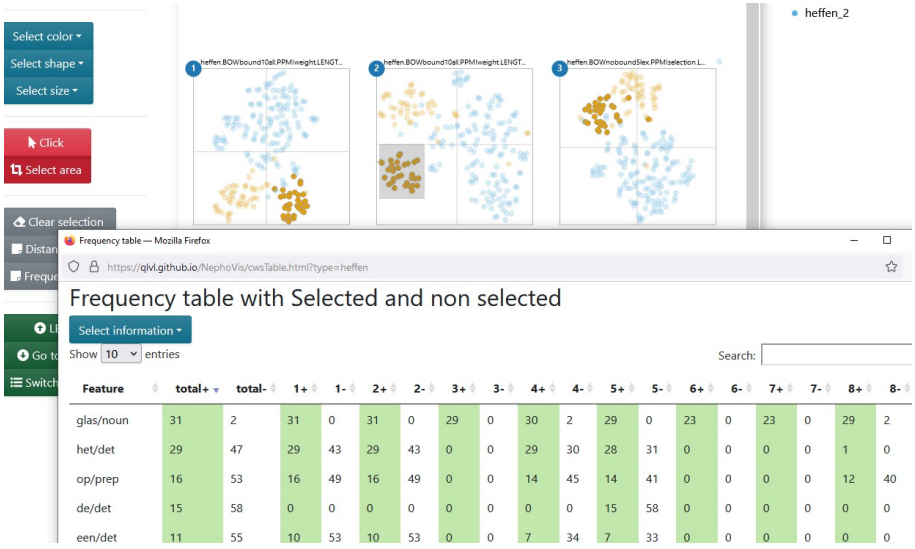

Figure 3.10: Level 2 for the medoids of *heffen* 'to levy/to lift', and frequency table of the the context words co-occurring with the selected tokens across models.

The layout of Level 2, showing multiple plots at the same time and linking the tokens across models, is a fruitful source of information, but it has its limits. To exploit more model-specific information, we go to Level 3.

3.4 Level 3

Level 3 of the visualization tool shows a zoomable, interactive scatterplot in which each glyph, by default a steel blue circle, represents a token, i.e. an attestation of the target lexical item. An optional second plot has been added to the right, in which each glyph, by default a steel blue star, represents a first-order context word, and the coordinates derive from applying the same dimensional-

ity reduction technique on the type-level cosine distances between the context words. The name of the model, coding the parameter settings, is indicated on the top, followed by information on the dimensionality reduction technique. Like in the other two levels, it is possible to map colours and shapes to categorical variables, e.g. sense labels, and sizes to numerical variables, e.g. number of available context words, and the legends are clickable, allowing the user to quickly select the items with a given value.

Figure 3.11 shows what Level 3 looks like if we access it by clicking on the name of the second model in Figure 3.9. Colour-coding and selection are transferred between the levels, so we can keep working on the same information if we wish to do so. Conversely, we could change the mappings and selections on Level 3, based on model-specific information, and then return to Level 2 (and refresh the page) to compare the result across models. For example, if the frequency table in Figure 3.10 had shown us that *glas* 'glass' was also captured in tokens outside our selection, or if we had reason to believe that not all of the selected tokens co-occurred with *glas* 'glass' in this model, we could input `glas/noun` on the `Features in model` field in order to select all the tokens for which *glas* 'glass' was captured in the model, and only those. Or maybe we would like to find the tokens in which *glasje* 'small glass' occurs, but we are not sure how they are lemmatized, so we can input `glasje` in the `Context words` field to find the tokens that include this word form in the concordance line, regardless of whether its lemma was captured by the model[10].

In sum, (groups of) tokens can be selected in different ways, either by searching words, inputting the id of the token, clicking on the glyphs or brushing over the plots.[11] Given such a selection, clicking on **Open frequency table** will call a pop-up table with one row per context word, a column indicating in how many of the selected tokens it occurs, and more columns with precomputed information such as PMI (see Figure 3.12). These values can be interesting if we would like to strengthen or weaken filters for a smarter selection of context words.

Like Level 2, Level 3 also offers the concordance line of a token when hovering over it. But unlike Level 2, the concordance can be tailored to the specific model on focus, as shown in Figure 3.11. The visualization tool itself does not generate a tailored concordance line for each model, but finds a column on the data frame that starts with `_ctxt` and matches the beginning of the name of the model to identify the relevant format. A similar system is used to find the appropriate list of context words captured by the model for each token. For these models, the selected context words are shown in boldface and, for `PPMIweight` models such as the one shown in Figure 3.11, their PPMI values with the target, e.g. *heffen*, are shown in superscript.

As we have seen along this chapter, the modelling pipeline returns a wealth of information that requires a complex visualization tool to make sense of it

[10] Admittedly, the names of the fields can be confusing and should probably be changed. Both fields work with partial regex matches, but `Features in model` look in the list of captured context words, which is a list of lemmas, while `Context words` performs the search on the concordance line, i.e. word forms, regardless of whether the model captured them.

[11] The (beta) feature of the type-level plot on the right side also enables token selection by clicking on the co-occurring context words (and *vice versa*) but this is still under development.

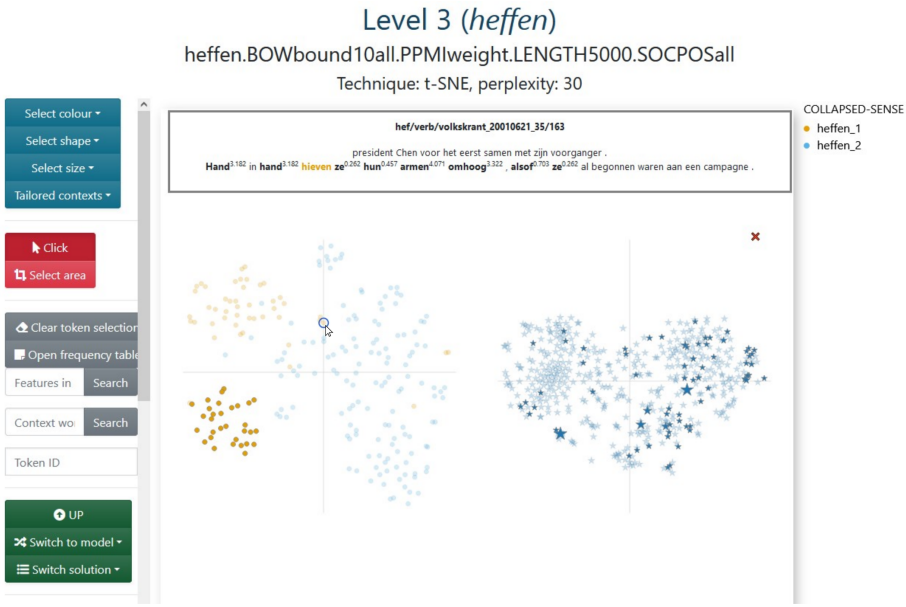

Figure 3.11: Level 3 for the second medoid of *heffen* 'to levy/to lift': bound10all-PPMIweight-5000all with some selected tokens. Hovering over a token shows tailored concordance line.

and exploit it efficiently. The Javascript tool described up to now, NephoVis, was developed and used by the same people within the Nephological Semantics projects, but is meant to be deployed to a much broader audience that could benefit from its multiple features. It can still grow, and its open-source code makes it possible for anyone to adapt it and develop it further. Nevertheless, for practicality reasons, an extension was developed in a different language: R. The dashboard described in the next section elaborates on some ideas originally thought for NephoVis and particularly tailored to explore the relationship between the t-SNE solutions and the HDBSCAN clusters on individual medoids.

3.5 ShinyApp

The visualization tool discussed in this section was written in R with the `shiny` library (Chang et al. 2021), which provides R functions that return HTML, CSS and Javascript for interactive web-based interfaces. The interactive plots have been rendered with `plotly` (Sievert et al. 2021). Unlike NephoVis, this tool requires an R server to run, so it is hosted on `shinyapps.io` instead of a static Github Page[12]. It takes the form of a dashboard, shown in Figure 3.13, with a few tabs, multiple boxes and dropdown menus to explore different lemmas and their medoids. All the functionalities are described in the About page of

[12]This code is also freely available at https://github.com/montesmariana/Level3.

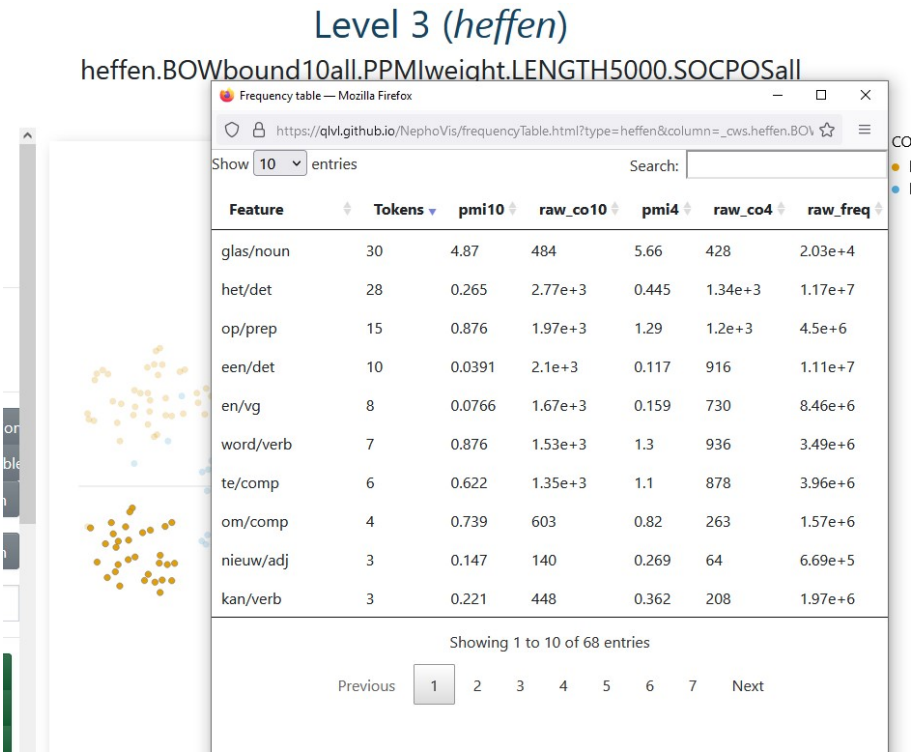

Figure 3.12: Level 3 for the second medoid of *heffen* 'to levy/to lift': bound10all-PPMIweight-5000all. The frequency table gives additional information on the context words co-occurring with the selected tokens.

the dashboard, so here only the most relevant features will be described and illustrated.

The sidebar of the dashboard offers a range of controls. Next to the choice between viewing the dashboard and reading the documentation, two dropdown menus offer the available lemmas and their medoids, by number. By selecting one, the full dashboard adapts to return the appropriate information, including the name of the model on the orange header on top. The bottom half of the sidebar gives us control over the definition of relevant context words in terms of minimum frequency, recall and precision, which will be explained below.

The main tab, **t-SNE**, contains four collapsable boxes: the blue ones focus on tokens while the green ones, on first-order context words. The top boxes (Figure 3.14) show t-SNE representations (perplexity 30) of tokens and their context words respectively, like we would find on Level 3 of NephoVis. However, the differences with NephoVis are crucial.

First, the colours match pre-computed HDBSCAN clusters ($minPts = 8$) and cannot be changed; in addition, the transparency of the tokens reflects their ε. The goal of this dashboard is, after all, to combine the 2D visualization

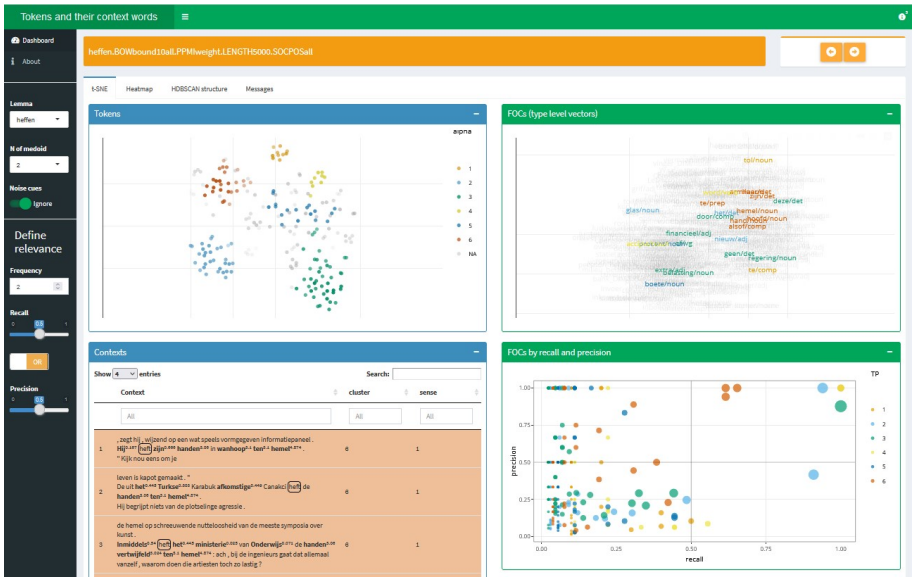

Figure 3.13: Starting view of the ShinyApp dashboard, extension of Level 3.

and the HDBSCAN clustering for a better understanding of the models. This functionality is not currently available in NephoVis because, unlike sense tags, it is a model-dependent categorical variable[13].

Second, the type-level plot does not use stars but the lemmas of the context words themselves. More importantly, they are matched to the HDBSCAN clusters based on the measures of frequency, precision and recall. In short, only context words that can be deemed relevant for the definition or characterization of a cluster are clearly visible and assigned the colour of the cluster they represent best; the rest of the context words are faded in the background. A radio button on the sidebar offers the option to highlight context words that are "relevant" for the noise tokens as well.

Third, the tooltips offer different information from NephoVis: the list of captured context words in the case of tokens, and the relevance measures as well as the nearest neighbours of the context word in the type-level plot. For example, in the left side of Figure 3.14 we see the same token-level model shown in Figure 3.11. Hovering over one of the tokens in the bottom left light blue cluster, we can see the list of context words that the model caputes for it: the same we could have seen in bold in the NephoVis rendering by hovering over the same token. Among them, *glas/noun* 'glass' is highlighted, because it is the only one that surpasses the relevance thresholds we have set. On the right side of the figure, i.e. the type-level plot we can see the similarities

[13]The current code is not suited to adapt the automatic selection of categorical variables to model-dependent ones, and adding the clustering solution for each of the models would clutter the list of categorical variables.

between the context words that surpass these thresholds for any cluster, and hovering on one of them provides us with additional information. In the case of *glas/noun* 'glass', the first line reports that it represents 31 tokens in the light blue HDBSCAN clusters, with a recall of 0.94, i.e. it co-occurs with 94% of the tokens in the cluster, and a precision of 1, i.e. it only co-occurs with tokens in that cluster. Below we see a list of the nearest neighbours, that is, the context words most similar to it at type-level and their cosine similarity. The fact that the similarity with its nearest neighbour is 0.77 (in a range from 0 to 1) is worrisome.

Figure 3.14: Top boxes of the **t-SNE** tab of the ShinyApp dashboard, with active tooltips.

The two bottom boxes of the tab show, respectively, the concordance lines with highlighted context words and information on cluster and sense, and a scatterplot mapping each context word to its precision, recall and frequency in each cluster. The darker lines inside the plot are a guide towards the threshold: in this case, relevant context words need to have minimum precision or recall of 0.5, but if they were modified the lines would move accordingly. The colours indicate the cluster the context word represents, and the size its frequency in it, also reported in the tooltip. Unlike in the type-level plot above, here we can see whether context words co-occur with tokens from different clusters. Figure 3.15 shows the right-side box next to the top token-level box. When one of its dots is clicked, the context words co-occurring with that context word — regardless of their cluster — will be highlighted in the token-level plot, and the table of concordance lines will be filtered to the same selection of tokens.

The first tab of this dashboard is an extremely useful tool to explore the HDBSCAN clusters, their (mis)match with the t-SNE representation and the role of the context words. In addition, the **HDBSCAN structure** tab provides information on the proportion of noise per medoid and the relationship between ε and sense distribution in each cluster. Finally, the **Heatmap** tab illustrates the type-level distances between the relevant context words, ordered

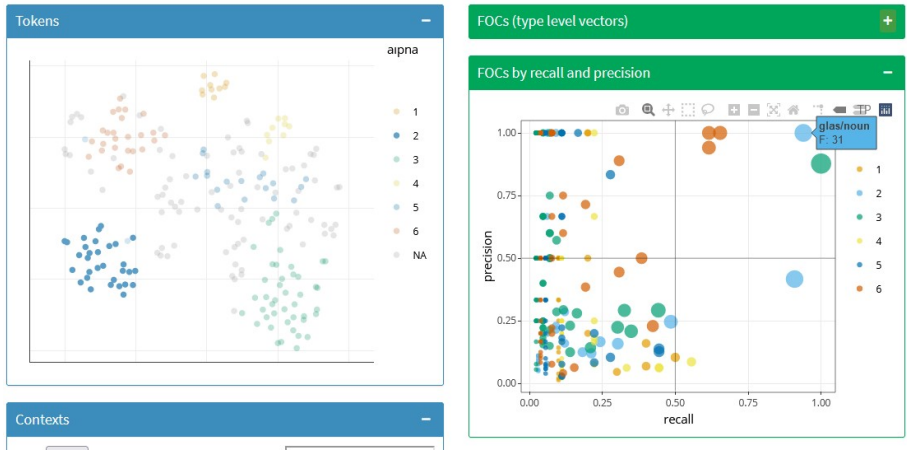

Figure 3.15: Token-level plot and bottom plot of context words in the **t-SNE** tab of the ShinyApp dashboard, with one context word selected.

and coloured by cluster, as shown in Figure 3.16. In some cases, it confirms the patterns found in the type-level plot; in others, like this model, it shows that most of the context words are extremely different from each other, forming no clear patterns. This is a typical result in `5000all` models like the one shown here and tends to lead to bad token-level models as well.

3.6 Summary

In this chapter two visualization tools for the exploration of token-level distributional models have been described. Both are open-source, web-based and interactive. They were developed within the Nephological Semantics projects at KU Leuven and constitute the backbone of the research described in this dissertation.

Data visualization can be beautiful and contribute to successful communication, but its main goal is to provide insight (Card, Mackinlay & Shneiderman 1999). Indeed, these tools have provided a valuable interface to an otherwise inscrutable mass of data. NephoVis offers an informative path from the organization of models to the organization of tokens, representing abstract differences generated by complicated algorithms as intuitive distances between points on a screen. Selecting different kinds of models and moving back and forth between different levels of granularity is just a click away and incorporates various sources of information simultaneously: find all models with window size of 5, look at them side by side, zoom in on the prettiest one, read a token, read the token next to it, find out its sense annotation, go back to the selection of models… Abstract corpus-based similarities between instances of a word, and between *ways* of representing these similarities (i.e. the models) become tangible, colourful clouds on a screen. Most of the points discussed in the sec-

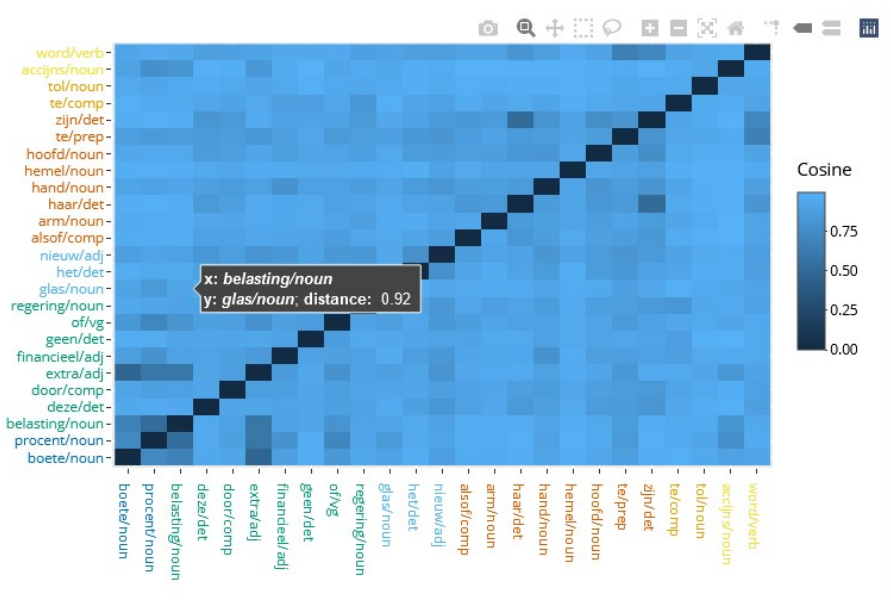

Figure 3.16: Heatmap of type-level distances between relevant context words in the ShinyApp dashboard.

ond part of this dissertation would have been simply impossible if it were not for these tools. Hopefully, they will prove at least half as valuable in future research projects.

Chapter 4

Case studies

Every empirical study needs a dataset. The methodological orientation of this project means that it does not aim for a linguistic description of some phenomenon in itself, but for the development of a tool that could aid such a description. Therefore, in order to test the workflow described in Chapter 2 and the visualization tools described in Chapter 3, the methodology was applied to a dataset. For that purpose, 32 Dutch nouns, adjectives and verbs exemplifying a range of semasiological phenomena were selected. The phenomena include: homonymy in the case of nouns, interaction between semantic variation and argument structure in the case of verbs and, for all parts of speech, metaphor, metonymy and generalization/specialization. The goal was to explore which phenomena were revealed by distributional models and whether they were related to certain parameter settings.

Homonymy occurs when the same lemma has two or more (sets of) senses that are not semantically or etymologically related. The rest of the relationships between senses can be broadly classified as generalization/specialization, metaphor, or metonymy. **Specialization** and **generalization** are two sides of the same coin: one of the senses involved is applied to a particular context or situation, and the other has a much broader application. Crucially, this process involves some additional semantic feature. For example, *herstructureren* 'to restructure' can be applied to a range of situations, but when it applies to companies or parts of companies in particular it does not only mean 'to change the structure of something' but also 'to reduce the personnel', which is missing in the general application. The direction of the relationship, i.e. whether the first sense is a generalization of the second or the other way around, is not relevant for the purposes of this study. The relevance is instead linked to the expectation that specialized senses would be more easily identified than general ones. Within Cognitive Linguistics, **metaphor** and **metonymy** are understood as cognitive principles that influence semantic structure, rather than mere expressive tools. They are found to interact and, at the same time, the distinction between them is not always unambiguous (Lakoff & Johnson 2003, Barcelona 2015, Lemmens 2015, Geeraerts 2003). While metaphor is described in terms of comparison, similarity and mapping between different

domains, metonymy is described in terms of reference, contiguity and mappings within a domain (Lemmens 2015). For example, when *grijs* 'gray' is applied to a weather-related term, e.g. *grijze avond* 'gray evening', the colour of the overcast sky stands for the weather in a metonymical mapping; when it is applied to an abstract entity like a *buurt* 'neighbourhood' a metaphorical sense 'boring, sad' is activated instead. However, the definition of what counts as a domain is not without problems, leaving the boundaries between metaphor and metonymy challenging to define as well (Croft 2003). For the purposes of these case studies, the distinction is relevant to the extent that metonymical senses are more likely than metaphorical senses to occur in the same contexts as their literal counterparts.

In practice, the situation is even more complicated. In the case of structural metaphors (Lakoff & Johnson 2003), metaphorical extensions might be elaborated by means of longer expressions. For example, in *we richten de spots op de zoektocht naar kandidaten* 'we aim the spotlights towards the search for candidates', *richten* 'to direct' and *op* 'on' can co-occur with either the literal or metaphorical senses of *spots* 'spotlight', and *zoektocht* 'search' is the cue that makes the literal sense less appropriate. This leads us to a situation already discussed by Geeraerts (2003) regarding the interaction of metaphor and metonymy in idiomatic and composite expressions. In a case like *hete aardappel* 'hot potato', which in the sample always refers to delicate situations that nobody wants to deal with, is the adjective 'hot' literal or metaphorical? Following Geeraerts' prismatic model of composite expressions, it could be explained as a combination of literal *heet* 'hot to the touch' with literal *aardappel* 'potato' that together is metaphorically understood as a delicate situation; a reinterpretation could then complete the mapping between the potato and the situation, and between the property of being hot to the touch and that of being delicate and to be avoided. The degree to which these reinterpreted mappings match systematic metaphorical or metonymical mappings of the individual elements is a separate issue: it could be argued for *heet*, which has a 'conflictive' meaning in non idiomatic constructions, but not for *aardappel* 'potato'. As a rule, these cases have been annotated as literal, understanding that it is the situation as a whole that is metaphorical.

It should be noted that these criteria are argumentative and justify the selection of the lemmas, but cannot go further than that. It is unfortunate, but the intriguing question about mapping parameter settings to these phenomena has a negative answer. As the second part of the dissertation will show, other factors play a role in the formation of the clouds, relegating these traditional semantic categories to a secondary place, if not as extras on the show. Nevertheless, the phenomena are accounted for, the questions have been asked and, no matter how unsatisfactorily, they have been answered.

Hence, this chapter focuses on the selection, collection and annotation of the dataset on which the methodology was tested. First, Section 4.1 will introduce the 32 selected lemmas and their senses, making explicit which of the aforementioned phenomena they exhibit. I will not discuss each lemma in detail; instead, I will expand of those used for illustration in Part II as it becomes relevant. Section 4.2 will focus on how the concordance lines were collected and

the manual annotation procedure. Relevant information regarding the annotation itself will also be provided. Finally, Section 4.3 rounds up the description and the technical part of this dissertation.

4.1 The lemmas

The selection of lemmas aimed to cover a wide range of phenomena: metaphor, metonymy, generalization/specialization, and more. The nouns were chosen because they exhibit both homonymy and polysemy: they have unrelated (groups of) meanings and at least one of them presents finer distinctions. The selection of adjectives also includes different kinds of semantic extension which are mostly related to the kind of noun that is modified by it. Finally, the verbs combine syntactic and semantic dimensions. The definitions provided to the annotators with their respective examples and their translations to English will be listed in tables, but no other examples will be shown in this chapter. Instead, relevant tokens and their contexts will be reproduced in the second part of the dissertation to illustrate the results from the analyses. Empty cells in the Dutch columns of the definitions indicate sense tags that were not present in the original selection of senses but instead were included *after* the annotation procedure — and assigned in a second stage — based on the results of the annotation itself. The Dutch definitions themselves are adaptations made by Dirk Geeraerts and me based on consultation of dictionaries (e.g. van Sterkenburg 1991, den Boon, Geeraerts & Arts 2007) and pilot surveys of small concordances from the corpus.

The selection of phenomena was attached to certain expectations. We expected specific senses to be easier to identify than general senses, i.e. to have a more identifiable context. With regard to nouns, homonyms were expected to be discriminated more easily that their internal distinctions. For verbs, instead, the expectation was to find more confusion between senses that either shared the semantic or the syntactic dimension than between senses that did not. We also expected metonymical senses to be harder to disambiguate than synaesthetic or metaphorical senses, since they are more likely to have an overlapping context with the more concrete, literal senses.

4.1.1 The nouns

A set of 7 nouns was selected that exhibit both homonymy and polysemy in at least one of the homonyms[1], as shown in Table 4.1. The purpose of this selection was to examine how models dealt with granularity, i.e. hierarchies of senses: homonyms should be easier to disambiguate than their senses, since they will apply to very different contexts, but maybe it would be possible to

[1] Originally we selected 8 nouns, but the very interesting *spoor* was discarded because of the high disagreement between the annotators and their (and my) difficulty understanding the definitions. It has three homonyms, 'trace', 'railway' and 'spurs', and some senses of 'trace' can be confused with some of 'railway'. In any case, the data is available for future analyses.

tune the parameter settings for different levels of granularity, like adjusting the focus on a camera.

Table 4.1: Definitions and examples for the senses of each of the 7 analysed nouns. In each sense, the first number indicates the homonym and, if there is a second number, the sense within the homonym.

Dutch	sense	English
	blik	
oogopslag (*een blik werpen op iets, een blik van verstandhouding*)	1.1	gaze (*throw a look at something, a look of understanding*)
gezichtsvermogen (*een scherpe blik*)	1.2	sight (*a sharp sight*)
inzicht, in intellectuele zin (*een brede blik*)	1.3	perspective, in intellectual sense (*a wide view*)
dun geplet metaal, i.h. bijz. vertind dun plaatstaal (*dozen uit blik*)	2.1	thin flattened metal, in particular thin tin-plated steel (*boxes of tin*)
voorwerp (i.h.bijz. doos voor voedsel) vervaardigd uit zulk materiaal (*stoffer en blik, een blik erwtjes, een maaltijd uit blik*)	2.2	object (in particular food container) made of tin (*brush and dustpan, a can of peas, canned meal*)
voedsel bewaard in een voorwerp als bedoeld in 2.2 (*eet je niet teveel blik?*)	2.3	food contained in an object as described by 2.2 (*don't you eat too much canned food?*)
	hoop	
ongeordende stapel (*een hoop rommel, gooi maar op de hoop*)	1.1	unordered mass (*a pile of junk, just drop it on the pile*)
grote hoeveelheid (*een hoop mensen, een hele hoop geld*)	1.2	great quantity (*a bunch of people, a lot of money*)
positieve verwachting, vertrouwen op iets positiefs (*hoop koesteren, de hoop uitspreken dat...*)	2	positive expectation, trust in something positive (*to nurture hope, express the hope that...*)
	horde	
bende, ordeloze groep personen (*een woeste horde*)	1	band, unordered group of people (*a ferocious horde*)
	1.2	unordered group of non-people (*a horde of computers*)
materiële hindernis, m.n. houten raamwerk gebruikt bij het hordelopen (*de 400m horden bij de vrouwen*)	2.1	material obstacle, namely wooden frames used for hurdling (*the 400m hurdles for women*)
hindernis in figuurlijke zin (*een horde nemen*)	2.2	obstacle in figurative sense (*to take a hurdle*)
	schaal	
een geordende reeks cijfers, afstanden, hoeveelheden e.d. waarmee iets gemeten wordt (*de schaal van Celsius, Richter, op een schaal van 1 tot 5*)	1.1	an ordered list of numbers, distances, quantities and such, with which something is measured (*the scale of Celsius, Richter, on a scale from 1 to 5*)
de verhouding tussen de grootte van iets en de weergave ervan in een kaart, model, grafiek etc. (*een schaal van 1:20, een schaal van 10 km*)	1.2	the ratio between the size of something and its representation in a map, model, graph etc. (*a scale of 1:20, a scale of 10km*)
grootteorde, omvang (*de schaal van een probleem, op grote/kleine schaal*)	1.3	magnitude, size (*the scale of a problem, on a large/small scale*)

Table 4.1: *(continued)*

Dutch	sense	English
harde buitenbekleding van zekere organische zaken (*de schaal van een ei, de schalen van een mossel*)	2.1	hard exterior of certain organic things (*the shell of an egg, the shell of a mussel*)
ondiepe en wijde schotel (*een schaal met vruchten*)	2.2	shallow and wide dish (*a platter with fruits*)
elk van de beide schotels die aan de armen van een balans hangen (*gewicht in de schaal leggen*)	2.3	each of the dishes hanging from the arms of a scale (*lay a weight on the (dish of a) scale*)
	spot	
	0	(idiosyncratic usage in sports headlines) (*Spot op 1ste*)
oneerbiedige, ridiculiserende uitspraak of behandeling (*de spot drijven met, bijtende spot*)	1	disrespectful, mocking expression or behaviour (*mock someone, sarcasm*)
reclameboodschap via radio, televisie, bioscoop (*een spotje voor tandpasta*)	2.1	advertisement via radio, television, cinema (*a spot for toothpaste*)
schijnwerper (*de spots richten op*)	2.2	spotlight (*direct the spotlights on*)
	2.3	metaphorical spotlight (*he likes to be in the spotlight*)
	staal	
zeer hard ijzer met laag koolstofgehalte (*twaalf ton staal, ijzer en staal, een man van staal*)	1.1	very hard iron with low carbon content (*twelve tons of steel, iron and steel, man of steel*)
	1.3	steel industry (*steel is striking*)
voorwerp of deel van een voorwerp uit zulk metaal (*het staal van de velgen is verroest*)	1.2	object or part of an object made of such metal (*the steel in the rims is rusted*)
monster van een stof of materiaal, bij wijze van proef (*een staal vragen, goederen op staal verkopen*)	2.1	sample of a substance or material, as evidence or proof (*to ask for a sample, to buy a sample of goods*)
proef, voorbeeld, bewijs (*een staaltje van hun kunnen, een staaltje van bewaamheid*)	2.2	proof, example, evidence (*a sample of their abilities, a proof of competence*)
	2.3	sample taken from a population for statistical analysis (*a representative sample*)
	stof	
materie, substantie van een bepaald type (*giftige stoffen, vaste stof, grijze stof*)	1.1	matter, substance of a certain kind (*poisonous substances, solid substances, gray matter*)
weefsel (*wollen en katoenen stoffen*)	1.2	fabrics (*woolen and cotton fabrics*)
onderwerp waarover men spreekt, schrijft, nadenkt etc. (*stof voor een roman, stof tot onenigheid*)	1.3	topic about which people talk, write, think, etc. (*material for a novel, topic of disagreement*)
massa zeer kleine droge deeltjes van verschillende oorsprong, door de lucht meegevoerd (*een wolk stof, stof afnemen*)	2.1	mass of very small dry particles of various origin, floating in the air (*a cloud of dust, to clean dust (=to dust)*)
massa zeer kleine deeltjes als toestand van een specifieke substantie (*iets tot stof vermalen, tot stof verpulveren*)	2.2	mass of very small particles as state of a specific substance (*to bring something to dust*)
	2.3	idiomatic uses of 'dust' (*lift up dust*)

Three nouns have one frequent, monosemous homonym and a less frequent, polysemous one: *hoop* 'hope/heap', *spot* 'ridicule/show or spotlight' and *horde* 'horde/hurdle'. The polysemy phenomena are varied. First, *horde* 'hurdle' can refer to literal hurdles, e.g. in races, while the other sense is metaphorical: abstract difficulties are talked about as obstacles to be surpassed. In addition, after the annotation a new sense tag derived from 'horde' was included, for the cases in which the members of the horde were not human beings, but insects, cars or other entities. Second, one of the *hoop* 'heap' senses refers to literal heaps of things that can form a pile, while the other one is a generalization to large quantities, e.g. *een hoop werk* 'a lot of work'. Finally, the polysemous homonym of *spot* has two main senses linked by metonymy, namely 'short video', e.g. and advertisement spot, or 'spotlight'. The 'spotlight' sense can also be used either literally or metaphorically ('to be in the spotlight'); this distinction was not included in the original definitions, but the annotators pointed it out and it was added afterwards.[2]

The other four nouns have two polysemous homonyms: *schaal* 'scale/dish', *blik* 'gaze/tin', *stof* 'substance/dust…', and *staal* 'steel/sample'. First, the frequent homonym of *blik* ('gaze') has a concrete sense with two metaphoric extensions: 'intellectual look', which was not attested in the sample, and 'perspective', which is quite infrequent. The infrequent homonym, 'tin', can either refer to the material itself, to an object made of that material ('tin can') or its content ('canned food'); due to their low frequency and the difficulty on part of the annotators to distinguish between the senses, the two last senses were later combined into one. Second, the frequent homonym of *stof* has two concrete, referentially distinct senses ('substance' and 'fabric') and an abstract one ('topic, material'). In contrast, for the less frequent homonym we distinguished two senses presenting a subtle, context-specific difference: between 'dust (in the air)' and the 'dust' in 'reducing something to dust, to pulverize'. The last sense was so infrequent that it was excluded, but another distinction emerged from the annotation, namely between literal 'dust' and 'dust' in idiomatic expressions, such as *stof doen opwaaien* 'to be controversial, lit. to stir up dust'. The new sense was added because, even though within the idiomatic expression the meaning of *stof* is still 'dust', the annotators kept confusing it with the 'topic, material' sense, which actually refers to expressions such as *stof voor een roman* 'material for a novel'. Third, *schaal* exhibits subtle perspective shifts in one homonym ('scale') and refers to different concrete objects with the second 'shell/dish', of which the very distinctive 'shell' sense was removed due to its low frequency. Finally, *staal* 'steel' could refer, like *blik* 'tin', to either the material or the part of an object that is made from it — the latter is very infrequent among our sample, but instead another sense could be identified, namely 'steel industry'. The 'sample' homonym, on the other hand, originally presented a metaphorical distinction between material samples and 'evidence' of abstract characteristics, but was modified after annotation to a specializa-

[2]Next to these more traditional tags, one category (sense 0 in Table 4.1) represents a selection of confusing lines with the format *Spots op {1ste, 2de, 3de}*. There are 13 such tokens in the sample, all from the start of Regional Sports articles from *Het Nieuwsblad*, published between 2003-09-10 and 2004-10-27. This category is not like any other sense, but it was included after the annotation just to see what the models did with it.

tion distinction between general samples, e.g. a urine sample, and (statistically) representative samples.

As we can see, the nouns present a variety of semantic phenomena at a finer granularity than homonymy: metaphor in the case of *blik* 'gaze', *horde* 'hurdle' and *spot* 'spotlight', metonymy in the case of *horde* 'horde', *blik* 'tin', *staal* 'steel' and *spot* 'videoclip/spotlight', generalization/specialization in the case of *staal* 'sample', *schaal* 'dish' and *hoop* 'heap', perspective shifts for *schaal* 'scale' and other relationships in the frequent *stof* homonym.

4.1.2 The adjectives

The selection of adjectives includes 13 lemmas presenting different kinds of polysemy phenomena (Table 4.2). The purpose of this selection was to examine how models dealt with their semantic relationships and whether they could extract them from the different nouns modified by the target adjective.

Three adjectives have a metonymic reading: *hoopvol* 'hopeful', *geestig* 'witty' and *hachelijk* 'dangerous/critical'. For *geestig* and *hoopvol*, one of the senses is anthropocentric, i.e. it's mainly or exclusively applied to people: witty people against the witty things they say or do, and people who express hope against things that inspire it. In *hachelijk*'s case, the difference is a matter of temporal or telic perspective: between things that might go wrong and situations that are already problematic.

Table 4.2: Definitions and examples for the senses of each of the 13 analysed adjectives.

Dutch	sense	English
	dof	
(van kleuren en zichtbare dingen) mat, zonder glans, vaal (*een doffe blik*)	1	(of colours and visible things) matte, without shine, pale (*a dull gaze*)
(van geluiden) niet luid of scherp, onderdrukt, gesmoord (*een doffe kreet*)	2	(of sounds) not loud or sharp, suppressed, smothered (*a dull cry*)
(van personen, gevoelens e.d.) niet opgewekt, lusteloos, zonder energie (*doffe onverschilligheid, doffe ellende*)	3	(of people, feelings, etc.) not cheerful, apathetic, without energy (*dull apathy, dull misery*)
(van denkbeelden e.d.) niet scherp voor de geest staand (*een doffe herinnering*)	4	(of ideas and such) not sharp in the mind (*a dull memory*)
	geestig	
scherpzinnig en humoristisch van aard (*een geestige collega*)	1	of witty and humoristic nature (*a witty colleague*)
blijk gevend van, uitdrukking gevend aan, gekenmerkt door scherpzinnigheid en humor (*een geestig boek, een geestige blik, een geestige opmerking*)	2	giving an impression of, expressing, characterized by wittiness and humor (*a witty book, a witty look, a witty remark*)
	3	being perceived as witty (*I find this funny*)

Table 4.2: *(continued)*

Dutch	sense	English
	gekleurd	
met kleur, in letterlijke zin (in het bijzonder, niet zwart, wit of grijs) (*gekleurde wangen*)	1	with colour, in a literal sense (in particular, not black, white or gray) (*colored cheeks*)
(van personen e.a.) niet blank (*de gekleurde medemens, van gekleurde afkomst zijn*)	2	(of people a.o.) not white (*the fellow colored man, to be of colored origin*)
(van uitspraken, opvattingen e.d.) niet neutraal, tendentieus (*een gekleurde voorstelling van zaken*)	3	(of expressions, concepts) not neutral, tendentious (*a colored representation of things*)
	geldig	
van kracht, van toepassing, van waarde zijnde volgens wettelijke of andere regels (*een geldig vervoerbewijs, betaalmiddel, juridisch bewijs*)	1	valid, acceptable, with value according to legal or other rules (*a valid driving license, currency, legal evidence*)
van kracht, van toepassing, van waarde in ruimere zin (*een geldige redenering*)	2	valid, acceptable, with value in general sense (*a valid reasoning*)
	gemeen	
gemeenschappelijk in gebruik of bezit, gedeeld (*gemene kosten, een gemene muur*)	1	common property or of common use, shared (*common costs, a common wall*)
openbaar, publiek (*de gemene zaak*)	2	public (*the public business*)
alledaags, gewoon, tot de middelmaat behorend (*het gemene volk, de gemene man*)	3	commonplace, normal, mediocre (*the common people, the common man*)
boosaardig, kwaadaardig, laaghartig, malicieus (*een gemene streek*)	4	malicious, evil, mean (*a mean trick*)
ordinair, plat, onkies, vulgair (*gemene praatjes*)	5	ordinary, flat, indecent, vulgar (*mean conversations*)
	6	cool, awesome, badass
	goedkoop	
laag in prijs, betaalbaar, voordelig (*goedkope wijn*)	1	of low price, affordable, advantageous (*cheap wine*)
geen hoge prijzen vragend (*een goedkoop winkeltje, een goedkope loodgieter*)	2	not asking a high price (*a cheap shop, a cheap plumber*)
waar de prijzen laag zijn (*een goedkope buurt*)	3	where the prices are low (*a cheap neighborhood*)
van weinig waarde, makkelijk verkregen, oppervlakkig, banaal (*goedkope lof, goedkoop succes, goedkope argumenten*)	4	with little value, received easily, superficial, banal (*cheap praise, cheap success, cheap arguments*)
	grijs	
met een kleur die ligt tussen wit en zwart; vaalwit, grauw (*grijs van het stof, de grijze dolfijn*)	1	with a color between white and black, pale white (*gray from the dust, the gray dolphin*)
(van periodes e.d.) zonder veel zonneschijn, bewolkt, betrokken (*een grijze dag*)	2	(of periods and such) without much sunlight, cloudy, covered (*a gray day*)

Table 4.2: *(continued)*

Dutch	sense	English
(van haar) zijn kleur verloren hebbend, m.n. door gevorderde leeftijd (*een grijs baardje*)	3	(of hair) having lost its color, namely because of old age (*a gray beard*)
(van personen e.a.) grijsharig, en vandaar, betrekking hebbend op ouderen (*de grijze golf*)	4	(of people and related) gray haired, and thus, related to old people (*the gray wave*)
saai, kleurloos, vervelend (*een grijze buurt*)	5	boring, not colorful, tedious (*a gray neighborhood*)
niet helemaal volgens de wet of de regels, halflegaal (*de grijze economie*)	6	not exactly following the law or rules, half legal (*the gray economy*)
hachelijk		
met kans op een ongunstige afloop, (potentieel) gevaarlijk (*een hachelijke onderneming*)	1	with chances of unfavorable outcome, (potentially) dangerous (*a dangerous enterprise*)
(reëel) gevaarlijk, netelig, kritiek, benard (*een hachelijke situatie*)	2	(actually) dangerous, trick, critical, dire (*a dangerous situation*)
heet		
(van dingen) zeer warm (*een gloeiend hete kachel*)	1	(of things) very warm (*a very hot stove*)
(van het lichaam) warm aanvoelend, een hogere temperatuur dan normaal hebbend (*hete wangen, het heet hebben*)	2	(of the body) feeling warm, having a higher temperature than normal (*hot cheeks, to feel hot*)
(van het weer) zeer warm (*hete dagen, hete zomer*)	3	(of the weather) very warm (*hot days, hot summer*)
(van voedsel) pikant (*hete sauzen*)	4	(of food) spicy (*hot sauce*)
(van personen) sexueel hartstochtelijk, geil (*een hete bok*)	5	(of people) sexually attractive, horny (*a hot buck*)
(van gebeurtenissen, periodes e.d.) gekenmerkt door heftige strijd (*het ging er heet aan toe, een hete herfst*)	6	(of events, periods, etc.) characterized by fierce conflict (*it was getting hot, a hot autumn*)
	7	popular, interesting or new, recent
heilzaam		
(letterlijk) bijdragend tot gezondheid en lichamelijk welzijn (*een heilzaam dieet*)	1	(lit.) that brings health and physical wellbeing (*a healthy diet*)
(figuurlijk) nuttig, een gunstig effect hebbend (*een heilzaam besluit*)	2	(fig.) necessary, having a beneficial effect (*a beneficial decision*)
hemels		
betrekking hebbend op de hemel (*de hemelse Vader, de hemelse boodschap*)	1	related to heaven (*de heavenly Father, the heavenly message*)
verrukkelijk, heerlijk, zalig, goddelijk (*een hemelse verschijning, een hemelse stem*)	2	delightful, lovely, blissful, divine (*a heavenly appearance, a heavenly voice*)
hoekig		
(van voorwerpen, figuren e.d.) met hoeken of scherpe kanten (*een hoekige tekening, een hoekig gezicht*)	1	(of objects, figures, etc.) with angles or sharp edges (*an angulous drawing, an angulous face*)
(van bewegingen, ritmes e.d.) niet vloeiend (*een hoekig melodietje*)	2	(of movements, rhythms, etc.) not fluent (*a broken melody*)

Table 4.2: *(continued)*

Dutch	sense	English
(van personen) houterig, stijf, onhandig in de omgang (*een hoekig karakter*)	3	(of people) rigid, stiff, clumsy (*a clumsy character*)
hoopvol		
(van personen, uitingen, gedragingen etc.) blijk gevend van hoop, vol hoop, optimistisch (*een hoopvolle stemming, dat stemt mij hoopvol*)	1	(of people, expressions, behaviors, etc.) giving an impression of hope, full of hope, optimistic (*a hopeful mood, that brings me hope (makes me hopeful)*)
reden tot hoop gevend, beloftevol (*hoopvolle perspectieven*)	2	giving reason for hope, promising (*hopeful perspectives*)

Four adjectives have metaphoric readings: *hoekig* 'angular', *dof* 'dull', *heilzaam* 'healthy/beneficial' and *gekleurd* 'colourful/person of colour/tainted'. *Heilzaam* has two senses, distinguishing between things that are literally healing, or beneficial for the health, and things that are metaphorically healing, or beneficial in general. *Hoekig* and *gekleurd* present three sense distinctions, one of which is particularly concrete and the most frequent, 'of angular form' and 'colourful' respectively, and another one explicitly anthropocentric: 'clumsy' and 'non white'. The third sense distinction has a different quality: synaesthetic for *hoekig*, applied to rhythms, and metaphoric for *gekleurd*, meaning 'tainted, corrupted'. Finally, *dof* has a concrete sense applied to the visual domain, a synaesthetic extension applied to sounds, and an abstract meaning applied to feelings and emotions; the fourth meaning listed in the table was not attested.

Three adjectives present some other form of similarity between the readings: *geldig* 'valid', *hemels* 'heavenly' and *gemeen* 'shared/mean…'. *Geldig* 'valid' and *hemels* 'heavenly' offer two options: one restricted to a specific context (laws and reglaments for *geldig* and Heaven for *hemels*) and one much broader. The case of *gemeen* is quite complex, involving a number of rather subtle distinctions that often co-exist in the same attestation: i.e. 'common' and 'shared', or 'average' and 'ordinary'.

Finally, the remaining three adjectives present a more complex picture: *heet* 'hot' and *goedkoop* 'cheap' have literal senses with different kinds of entities but has also metaphorical extensions, while *grijs* 'grey' has both metaphorical and metonymical extensions. *Heet* 'hot' presents, first, three very concrete senses that differ in perspective: temperatures of objects, of weather and as it is felt in the body; the other three senses are metaphorical, i.e. the objects to which *heet* is applied cannot be physically hot. Crucially, there is no exclusive sense tag for idiomatic expressions in which the combination of *heet* 'hot' and its concrete object (e.g. *hang_ijzer* 'iron', *aardappel* 'potato') is used metaphorically. *Goedkoop*, on the other hand, presents a modest set of 4 sense distinctions: a concrete, prototypical and frequent sense (i.e. cheap products), two perspectival shifts (i.e. cheap shops and cheap area) and a clear metaphor (i.e. of little values). Finally, *grijs* presents a very frequent, concrete sense, three specific

metonymic extensions — to weather and to hair, and from there to old people or generations — and two metaphorical ones — 'boring' and 'half legal'. In practice, the 'boring' reading can include 'sad, not cheerful', and the 'half legal' sense is more general, applying to 'gray areas' between two poles.

In sum, the adjectives include more simple semasiological structures with only one kind of semantic extension involved as well as more complex interactions between the phenomena.

4.1.3 The verbs

The criterion to select the 12 verbs analysed here was to cover a range of combinations of syntactic and semantic variation, with the goal of exploring how different parameter settings dealt with their interaction or whether certain types of models would focus on one or the other aspect.[3] Their senses and translations are shown in Table 4.3.

Four verbs are always transitive and their senses can be distinguished by the objects they can take: people or objects for *haten* 'to hate', people or opinions for *huldigen* 'to honour/to believe', concrete objects or taxes for *heffen* 'to levy/to lift', and statements or decisions for *herroepen* 'to recant/to void'.

Two of the verbs can be transitive, with a distinction based on the direct object, or intransitive: *helpen* 'to help' and *herstructureren* 'to restructure'. In both cases the intransitive sense is semantically similar to one of the transitive senses. For example, the intransitive sense and one of the transitive senses of *herstructureren* only apply to companies, with the connotation that the personnel is being reduced, while the other transitive sense has a much more general application.

Three verbs can be transitive, with a distinction based on the direct object, or reflexive: *diskwalificeren* 'to disqualify', *herhalen* 'to repeat' and *herinneren* 'to remember/to remind'. In the case of *diskwalificeren* 'to disqualify' and, to a lesser degree, *herhalen* 'to repeat', this opposition can be interpreted as a specific situation where the object and the subject coincide. In contrast, *herinneren* means 'to remember' in the reflexive construction and 'to remind' in the transitive construction with the preposition *aan*; the transitive construction without the preposition can also be attested (e.g. *ik word herinnered als*, 'I am remembered as') but very infrequently.

Table 4.3: Definitions and examples for the senses of each of the 12 analysed verbs.

Dutch	sense	English
diskwalificeren		
(trans.) ongeschikt verklaren en uitsluiten van een bepaalde functie of positie (*een getuige diskwalificeren*)	1	(trans.) declare unsuitable and exclude from a certain function or position (*disqualify a witness*)

[3]The original set of verbs also included *herkennen*, but it was excluded because of the extreme subtlety of its sense distinctions, which made the annotation particularly challenging.

Table 4.3: *(continued)*

Dutch	sense	English
(trans.) wegens onregelmatigheden uitsluiten bij een wedstrijd (*FC De Trappers werd gediskwalificeerd wegens wangedrag*)	2	(trans.) exclude from a competition because of irregularities (*FC De Trappers was disqualified because of misbehaviour*)
(reflex.) zichzelf buiten spel zetten, zich onmogelijk maken (*met zulk gedrag diskwalificeer je jezelf*)	3	(reflex.) exclude oneself, make oneself impossible (*with such a behaviour you disqualify yourself*)
	haken	
(trans.) met of als met een haak vastmaken (aan, in, achter iets) (*een wagen aan een locomotief haken, een sleutel in een ring haken*)	1	(trans.) fix something with or as if with a hook (at, to, behind something) (*hook a wagon to a locomotive, a key in a key ring*)
(intrans.) met of als met een haak vastraken (*de doornen haakten aan haar jas, haar paraplu bleef haken aan de deurknop*)	2	(intrans.) get stuck with or as if with a hook (*the thorns got stuck in her coat, her umbrella got stuck in the doorknob*)
(trans.) over een uitgestoken been doen struikelen (*hij werd gehaakt in de elfmeter, iemand pootje haken*)	3	(trans.) make trip over a stuck out leg (*he was made to trip in the penalty kick, make someone trip*)
(intrans., met 'blijven') van gedachten, blikken e.d.: haperen, telkens terugkeren (aan of bij iets) (*ik bleef haken bij de herinnering aan mijn broer*)	4	(intrans., with 'to keep') of thoughts, gazes and such: falter, come back (to something) (*I kept going back to the memory of my brother*)
(intrans./trans.) zeker handwerk maken door met een staafje met een weerhaak lussen samen te weven (*haken tijdens het televisiekijken, hoe ontspannend!, een babymutsje haken*)	5	(intrans./trans.) make handcraft by weaving loops together with a hooked needle (*crochetting while watching tv, so relaxing!, crochet a baby hat*)
	6	(with 'towards') desire, aim for
	harden	
(trans.) hard maken, in letterlijke zin (*staal harden*)	1	(trans.) make hard, in literal sense (*harden steel*)
(intrans.) hard worden, in letterlijke zin (*snel hardende verven*)	2	(intr.) become hard, in literal sense (*quickly hardening paint*)
(trans.) hard maken in figuurlijke zin; weerstand en veerkracht bijbrengen (*een kind harden tegen het klimaat*)	3	(trans.) make hard in figurative sense; impart resistance and resilience (*toughen a child against the weather*)
(reflex.) bij zichzelf weerstand en veerkracht aankweken (*zich harden tegen het lot*)	4	(reflex.) develop resistance and resilience by oneself (*toughen oneself against fate*)
(trans.) uithouden, verdragen (*niet te harden*)	5	(trans.) endure, tolerate (*unbearable ('not to bear')*)
	haten	
(trans.) iem. haat toedragen, een sterk gevoel van afkeer en vijandschap t.o.v. iem. hebben (*waarom haat hij mij zo?*)	1	(trans.) feel hatred, have a strong feeling of aversion and enmity towards someone (*why does he hate me so much?*)
(trans.) iets onaangenaam, verfoeilijk, verwerpelijk vinden (*hoe zou iemand de taalkunde kunnen haten?*)	2	(trans.) consider something unpleasant, detestable, reprehensible (*how could someone hate linguistics?*)

Table 4.3: *(continued)*

Dutch	sense	English
	heffen	
(trans.) m.b.t. materiële zaken: in de hoogte brengen, optillen (*met geheven hoofd; hij heft met gemak 80 kilo in de hoogte*)	1	(trans.) w.r.t. material objects: move to a higher position, lift (*lifting their head; he easily lifted 80 kg*)
(trans.) m.b.t. geld e.d.: invorderen, eisen, opleggen (*belasting, rente, accijns heffen*)	2	(trans.) w.r.t. money and such: collect, demand, impose (*collect tax, interest, excise*)
	helpen	
(trans.) ondersteunen in materiële of morele zin, bijstaan (*met raad en daad helpen, een helpende hand, uit de nood helpen*)	1	(trans.) support in material or moral sense, assist (*help in word and deed, a helping hand, help out*)
(trans.) iem. assisteren door met hem samen te werken (*helpen met het huiswerk; heb je dat alleen gedaan of heeft iemand je geholpen?*)	2	(trans.) assist someone by collaborating with them (*help with homework, did you do that by yourself or did someone help you?*)
(intrans.) voordeel opleveren, nuttig zijn (*dat drankje heeft goed geholpen*)	3	(intrans.) yield advantage, be useful (*that drink helped a lot*)
	4	(trans.) with inanimate entities, be helpful, useful
	5	(with 'to/for') to provide
	herhalen	
(trans.) m.b.t. handelingen of activiteiten: opnieuw uitvoeren (*een experiment, een les, een bezoek herhalen*)	1	(trans.) w.r.t. acts or activities: perform again (*repeat an experiment, a lesson, a visit*)
(trans.) m.b.t. zinnen, boodschappen e.d.: opnieuw uitspreken (*kunt u dat even herhalen?*)	2	(trans.) w.r.t. utterances, messages and such: pronounce again (*Could you please repeat that?*)
(reflex.) zich opnieuw voordoen (*de geschiedenis herhaalt zich*)	3	(reflex.) occur again (*history repeats itself*)
	4	(trans.) of a show or an episode, broadcast again
	herinneren	
(met 'aan') weer te binnen brengen, in het geheugen terugroepen (*iemand aan iets herinneren*)	1	(with 'of') bring back to the mind, to the memory (*remind someone of something*)
(reflex.) in het geheugen aanwezig hebben, niet vergeten (*zich een gebeurtenis, een persoon herinneren*)	2	(reflex.) have present in the memory, not forget (*remember an event, a person*)
(trans.) met een plechtigheid, monument o.i.d. gedenken (*we herinneren vandaag de Slag bij Ronceval*)	3	(trans.) remember with a celebration, monument and such (*today we remember the Battle of Roncevaux Pass*)
	herroepen	
(trans.) m.b.t. wetten, besluiten e.d.: intrekken, niet langer geldig verklaren (*een besluit, volmacht, decreet herroepen*)	1	(trans.) w.r.t. laws, decisions and such: withdraw, declare not valid anymore (*annul a decision, power of attorney, decree*)

Table 4.3: *(continued)*

Dutch	sense	English
(trans.) m.b.t. uitspraken, meningen e.d.: terugnemen en rechtzetten (*Trump moest weer een van zijn dwaze tweets herroepen*)	2	(trans.) w.r.t. statements, opinions and such: retract and correct (*Trump had to retract one of his crazy tweets again*)
herstellen		
(trans.) repareren, de eraan ontstane schade wegwerken (*het dak herstellen*)	1	(trans.) repair, get rid of the damage in something (*repair the roof*)
(trans.) tot de vorige toestand terugbrengen, doen terugkeren (*de goede verstandhouding herstellen*)	2	(trans.) bring back, make return to the previous state (*repair the understanding*)
(trans.) goedmaken, weer doen vergeten (*een fout herstellen*)	3	(trans.) make good, make forget (*fix a mistake*)
(reflex.) tot de oorspronkelijke toestand terugkeren (*de rust herstelt zich*)	4	(reflex.) return to the original state (*peace is restored*)
(intrans.) genezen (*van een ziekte herstellen*)	5	(intrans.) heal (*heal from a disease*)
	6	(intrans.) of a financial/economic entity, recover
herstructureren		
(trans.) reorganiseren, een nieuwe structuur geven (*je kunt deze tekst maar beter herstructureren*)	1	(trans.) reorganize, give a new structure (*you should restructure this text*)
(trans.) m.b.t. bedrijven in problemen: activiteiten of personeel afstoten, downsizen (*Bayer herstructureert zijn plasticdivisie*)	2	(trans.) w.r.t. businesses in difficulties: remove activities or personeel, downsize (*Bayer restructures its plastic division*)
(intrans.) van bedrijven in problemen: activiteiten of personeel afstoten, downsizen (*de chemie moet zich herstructureren*)	3	(intrans.) of businesses in difficulties: remove activities or personeel, downsize (*chemistry must restructure (itself)*)
huldigen		
(trans.) iets of iem. eer bewijzen, vieren (*we huldigen de uitvinder van de herbruikbare broodzak*)	1	(trans.) celebrate, pay homage to someone or something (*we honor the inventor of the reusable bread bag*)
(trans.) erkennen, aankleven, toegedaan zijn (*een opvatting, mening, theorie huldigen*)	2	(trans.) acknowledge, follow, be commited to (*hold a view, an opinion, a theory*)

Two more verbs can be transitive, intransitive or reflexive, with semantic distinctions within the transitive structure: *harden* 'to make or become hard/ to tolerate' and *herstellen* 'to repair/ to heal…'. The senses of *harden* can be split in two main groups. One is more closely related to the property of 'hardness', i.e. to turn something or someone hard or to become hard, in literal or figurative sense, with different constructions: from the intransitive literal sense in *om hun kaas te laten harden* 'in order to make their cheese harden' to the transtive figurative one in *Verdriet heeft haar gehard* 'Grief has hardened her'. The second group, however, includes one transitive construction in a very specific pattern but is more frequent in the sample than all the others combined:

(niet) te harden ('to (not) tolerate', always negative).

Finally, *haken* 'to hook' presents semantic distinctions within both the transitive and the intransitive structures. It can refer literally or metaphorically to hooking something or remaining hooked, but there are also two very specific senses: one characteristic of the football context, meaning 'to make someone trip (by placing a foot in front of them)', and 'to crochet'.

In sum, the set of verbs includes cases where only the kind of direct object plays a role in the disambiguation and cases where it interacts with syntactic patterns. Moreover, the specific ways in which these kinds of direct objects are defined differ across verbs: from animacy or agency in the case of *haten* to concreteness in the case of *heffen*. The semantic distinctions can also rely on a broader context: *diskwalificeren* will typically have people as direct object, but the sports-related context defines a specific sense, characterised by distinct motivations and consequences.

4.2 The dataset

For each of the 32 lemmas listed above, about 300 tokens were collected from the *QLVLNewsCorpus* (described in Section 2.3.1). All attestations were manually annotated by at least three different people based on the definitions found in the Dutch column of Tables 4.1, 4.2 and 4.3. Next to the sense assignment, which was later revised for uniformity — and to include senses emerging from the annotation itself, as mentioned above — the annotation included confidence assignment and selection of disambiguating context words.

The selection of the lemmas involved some introspection as well as consultation of lexical resources and corpus data: thinking of potential candidates, checking the senses reported in dictionaries (van Sterkenburg 1991, den Boon, Geeraerts & Arts 2007) and estimating their relative frequencies in small concordances. We tried to avoid extremely skewed distributions approximating a monosemous structure or numerous infrequent senses that would be unlikely to stand out in a model.[4] In the end, as we will see, sense frequency is not really an issue, because clouds don't model senses anyways.

The exploration of these samples of concordances also served for the calculation of the number of tokens to model and annotate. Regardless of the actual frequency of the items in the corpus, the minimum sample contained 240 tokens; it was raised to 280 if any of the senses had a relative frequency below 20% in the sample, to 320 if it was below 10%, and to 360 if there were many senses and therefore some had a low frequency (e.g. *heet*). The lower and upper bound were estimated from pilot studies of clouds as a large enough amount to warrant the use of this methodology and small enough to make sense of in the visualization tool. Table 4.4 shows the absolute frequency (in

[4]In a number of cases, the corpus survey (reading a random concordance of 40-50 lines) invalidated options that intuitively or according to the dictionary definitions would have conformed to our requirements. When judging such a discrepancy, it is important to take into account the composition of the corpus. The topics addressed in newspapers and the terms used to talk about them are certainly not representative of everyday life or the entirety of language.

the 520MW *QLVLNewsCorpus*) of each selected lemma, the size of the sample and the distribution of the senses: the more the boxplot in the rightmost column goes to the right, the more frequent one of the senses. For example, the long boxplots for *blik* and *hoop* indicate a very skewed distribution, i.e. a sense with very high frequency and senses with very low frequencies, while the narrow, centred boxplots for *hachelijk* and *hemels* indicate that their senses are equally frequent. The sample extraction was almost completely random, with the only restriction that no two instances of the same lemma would be extracted from the same file. There were, however, a few duplicates, due to repetition of the same fragment on different dates.

Table 4.4: Absolute frequency of the lemmas in the corpus, number of batches and distribution of their senses. The number next to the boxplots indicate the number of different senses.

lemma	frequency	sample	senses
nouns			
spot	3496	240	5 
horde	3224	280	4 
blik	22175	280	4 
staal	5796	320	5 
schaal	14249	320	5 
stof	24502	320	5 
hoop	41946	320	3 
adjectives			
hachelijk	1307	240	2 
hemels	1417	240	2
heilzaam	1476	240	2 
hoopvol	3680	240	2 
geldig	5128	240	2 
hoekig	1242	280	3 
geestig	3970	280	3 
gekleurd	4520	280	3
dof	1268	320	4
gemeen	2997	320	7 
grijs	13567	320	7 
goedkoop	40669	320	4 
heet	10676	360	7
verbs			

Table 4.4: *(continued)*

lemma	frequency	sample	senses
herroepen	848	240	2
herstructureren	936	240	3
diskwalificeren	1084	240	3
huldigen	4091	240	2
heffen	4799	240	2
haten	4828	240	3
herstellen	28814	240	6
herinneren	33432	240	3
helpen	87136	240	6
harden	1050	320	5
herhalen	16856	320	4
haken	1403	360	6

For each of the tokens a concordance line was extracted with 15 words to either side. Bachelor students of Linguistics at KU Leuven were recruited and hired to manually annotate the samples of the selected lemmas. Each of them was tasked with annotating 40 tokens of each of 12 types (at least three nouns, four adjectives and four verbs, plus one of either of the categories)[5]: a total of 480 tokens[6], to annotate in 6 weeks. In total, each of the 9600 tokens was annotated by at least three annotators; 10% of them were annotated by four. Each lemma was split in 6-9 batches of 40 tokens, each of them annotated by a different group of annotators. The annotators were offered an introductory meeting, a video tutorial and written guidelines, but the procedure itself was performed individually.

Both the lemmas and the batches were assigned randomly, while keeping in mind the part-of-speech distribution. It was the intention to shuffle the samples of each lemma before splitting them into batches, but something went wrong with the code and they were ordered by source; each batch would have mostly tokens of a different newspaper. The annotation involved three tasks:

1. Assign a sense from a predefined set of definitions, namely the Dutch column in Tables 4.1 through 4.3. If none of the tags apply, select "None of the above" and explain why;
2. Express the confidence of the decision in a scale of 6 values;
3. Identify the words of the context that helped in the disambiguation.

Since entering textual information in a spreadsheet can easily lead to typos and inconsistencies and, furthermore, annotating the helpful context words

[5]Recall that originally there were 8 nouns and 12 verbs.
[6]A few of them doubled their load and annotated two sets of 480 tokens.

is particularly challenging in such a tool, a user-friendly visual interface was designed that received input from buttons and returned the output in JSON format. The interface, which is not available in its original form any more, had a menu with the list of lemmas and two tabs: an overview of the concordance lines of the selected type and an annotation workspace (Figure 4.1). The annotation workspace focused on one concordance line[7] (or token) at a time, offering first the text, then a series of long radio buttons with the definitions and examples, a star rating option for the confidence evaluation, followed by a clickable reproduction of the text, and a text input field for comments. The long radio buttons meant that the annotators had the full definitions and examples at their disposal every time they had to assign a sense for a given lemma, while the final output transformed their decisions into more manageable codes, such as `sense_1`, `sense_2`, etc. The clickable concordance lines let them select the context words they deemed most useful to the annotation procedure by simply clicking on them; the program then translated this as an array of positions relative to the target, e.g. `["R1", "L2"]` if the first word to the right and the second to the left are selected.[8] Finally, the text input field at the bottom was available to leave any sort of comment and was compulsory when "None of the above" was selected.

The dataset obtained from this procedure is very rich and interesting for a variety of purposes. For each token we have sense assignment, confidence evaluation and selection of informative cues by at least three different independent annotators, as well as comments on at least the cases which did not receive a sense. Agreement between the annotators can be measured with coefficients such as Fleiss' κ (Fleiss 1971), illustrated in Figure 4.2, but the resulting picture may be unnecessarily complex. First, disagreement is susceptible to granularity: annotators might disagree between senses of a noun but not between the homonyms, except for their confusion between idiomatic senses of *stof* 'dust' and its 'topic, material' sense. Second, annotators were not very sensitive to grammatical distinctions (e.g. between transitive and intransitive senses), which was a strong reason for disagreement in *herstructureren*, *helpen*, *haken* and *herstellen*. Third, disagreements were sometimes concentrated on one annotator, who showed a strong preference for a certain sense; as such, they were not an indicator of the ambiguity of the token but of misunderstandings on the part of the annotator. Some annotators exhibited an almost excessive attention to nuances, while others were much less thorough.

More importantly, for the great majority of the tokens (83.8%) the majority of the annotators agreed on one tag that remained as the official sense for that

[7] The web-based interface interpreted HTML, of course. As a consequence, the sentence separator `<sentence></sentence>` was simply ignored; it should have been replaced with `<p></p>` to properly render the division, especially after headlines, which lack final stops. Interestingly, this rendering also could have read strings such as `&quot;` as `"`, but at some earlier stage of pre-processing of the corpus all the `&` have been transformed into `and`, resulting in a number of confusing appearances of `andquot;` in the concordance lines.

[8] For a few weeks into the annotation, the code had a bug that meant that if a word form was repeated in the concordance line and one of its instances was selected its first occurrence was recorded even if the chosen one was a later co-occurrence. The bug was fixed as soon as it was reported and the rest of the annotators were warned, but not all the resulting errors were corrected.

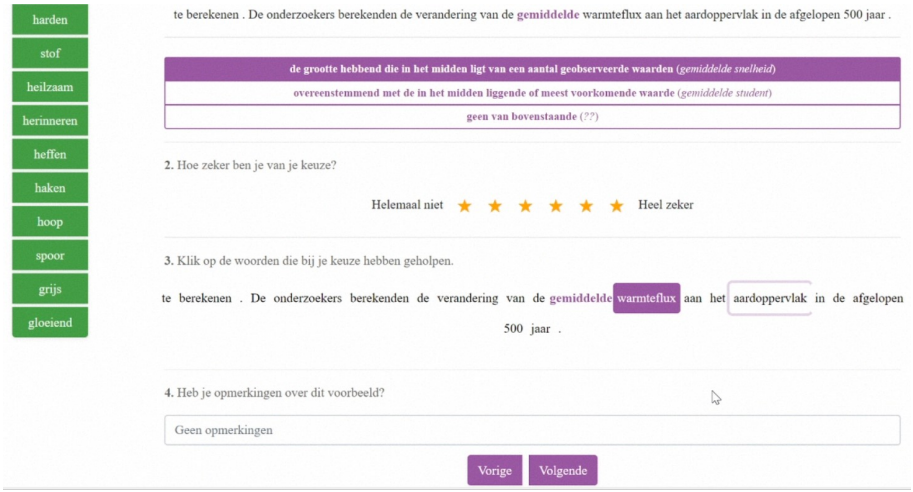

Figure 4.1: Screenshot of the options in the annotation tool.

token. After gathering and exploring the data, the tokens were reread by me and a final decision was made for their sense tags. Figure 4.3 shows the number of tokens with full agreement, a majority agreement (i.e. only one annotator disagreed) or no agreement and whether the same chosen sense was kept in the final annotation, another tag was applied or the token was removed (e.g. tokens of *heet* that corresponded to the verb *heten*). The `Other` category includes new senses suggested by the annotators themselves as well as corrections from misunderstandings, such as the second original sense of *blik*, which annotators interpreted in different ways and was actually not attested in the dataset. The very few cases of `Same` with no agreement were tokens annotated by four annotators where two of them selected the senses that remained, while the other two disagreed.

In addition, the final sense distribution is not significantly different from that in the much smaller pilot samples. Distribution across batches, instead, was affected by regional variation. For example, Belgian sources include more sports-related articles than the Netherlandic sources, leading to variation in the sense distribution of lemmas with such a sense (*diskwalificeren* 'to disqualify', *haken* 'to make someone trip' and *horde* 'hurdle') across regions. This discrepancy in distribution across batches could have been avoided if the tokens had been properly shuffled.

Around 4% of all the assigned tags where "None of the above", with a clearly uneven distribution. The lemmas with the largest amount of were *haken*, with 117 tokens in which three annotators chose "None of the above" and 72 in which two of them did. *Heet* and *harden* follow with 69 and 90 tokens with 3 such tags and 14 and 10 with two. Many of these were due to wrong lemmatization: the concordance of *haken* had many instances of *afhaken* 'to stop' or *met haken en ogen*, an idiomatic expression in which it is a noun; the concordances of

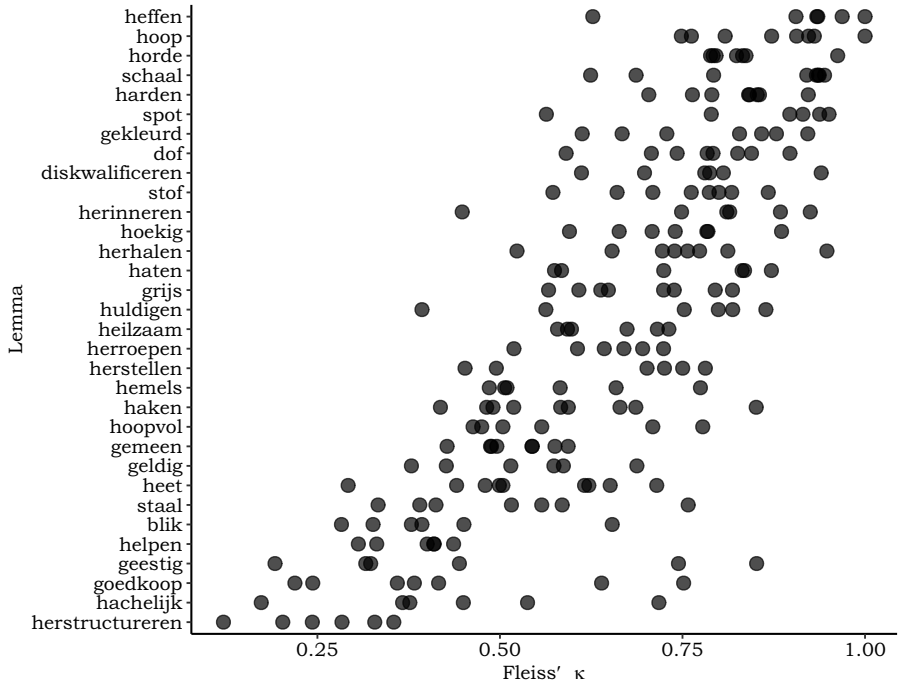

Figure 4.2: Agreement between annotators per batch per lemma, computed with `irr::kappam.fleiss()` (Gamer et al. 2019).

heet and *harden* included instances of the verb *heten* 'to call, to be named' and the adjective *hard* respectively. In a similar way, many of the tokens in the concordance of *heffen* were instances of *opheffen* 'to lift/to cancel', but the annotators did not always catch these cases. The verbs *afhaken* and *opheffen* are separable verbs in Dutch: in some constructions, the prefix is separated from the root, so that a syntactic parser might confuse them with a different verb and a preposition. Next to these issues, annotators assigned "None of the above" in cases where the tokens did not match any of the suggested senses, especially in cases of idiomatic expressions such as *hete aardappel* 'hot potato'. All these annotations where classified in four categories: `wrong_lemma`, for the cases of wrongly selected concordance lines, was assigned to 413; `not_listed`, assigned 421 times, indicated that the lemma was correct but none of the suggestions applied; `unclear` (240 times) was used when the token could not be parsed by the annotator, and `between` (45 cases) referred to doubt between two or more of the given options. These different classes informed later decisions such as whether to add or remove senses or tokens.

Tokens were removed for different reasons. Next to the cases where the concordance line did not belong to begin with (including adverbial uses of the adjectives), there were some indecipherable tokens, extremely infrequent senses

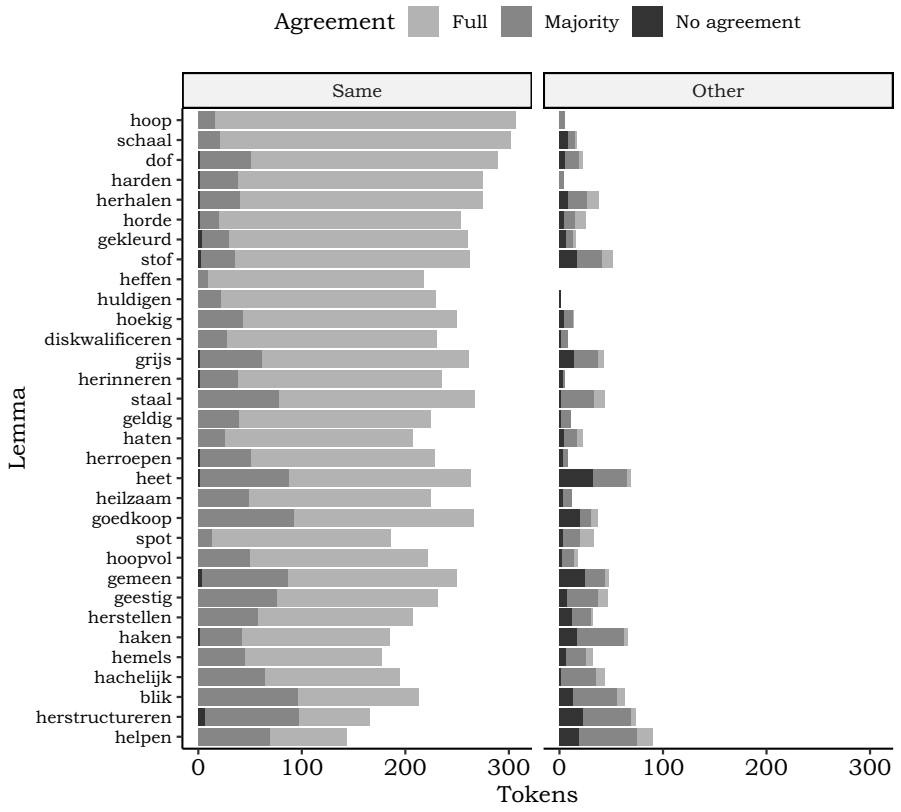

Figure 4.3: Number of tokens per lemma with full, partial (majority) or no agreement, split by whether the majority sense was kept or changed. Removed tokens are not included.

(e.g. 4, 5 tokens out of 250) and duplicated tokens. In total, 424 tokens were removed, 109 of which belonged to *haken*.

Confidence values were explored but not used, because they tend to be similar across batches, lemmas and senses, with a tendency towards the highest values and variation across annotators instead: what is low confidence for some of them is high confidence for others. Figure 4.4 breaks this down in terms of the degree of agreement and whether the assigned tag matched one of the senses offered or not. Note that the top facet, "None of the above", has much lower counts than the lower facet. We would expect confidence ratings to be lower for annotations that do not agree with the other votes for the same token and, in relative terms, that is the case. Confidence assignment to a "None of the above" tag is ambiguous: some annotators tend to give them the minimum confidence because they are not confident about the meaning of the concordance line, while others assign a high value because they are confident that none of the

other options applies.

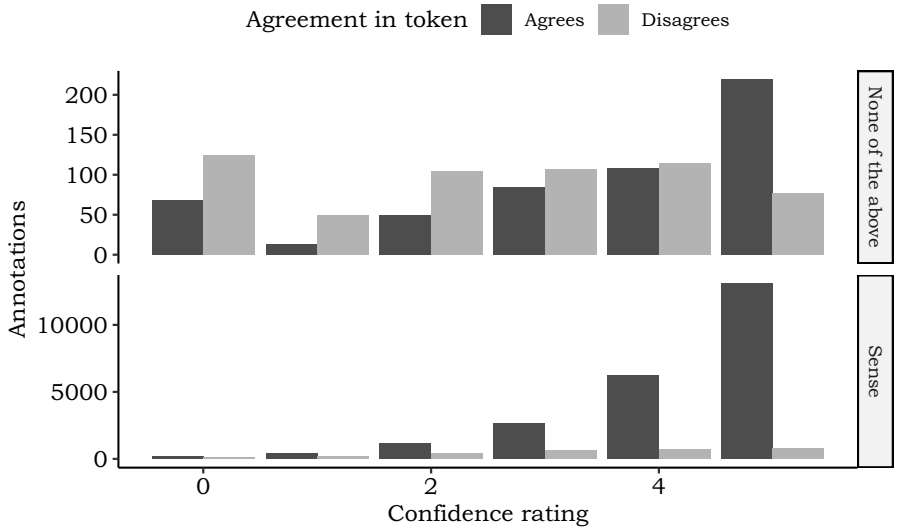

Figure 4.4: Distribution of confidence values across annotations, by whether the annotators agreed with another in the same token and by whether they selected a sense or "None of the above".

The selection of cues was consulted when defining parameter settings (Section 2.3): if two annotators agreed on both the sense tag and a context word for a given token, that context word was considered an official **cue** for that sense. From the relative position representing the cue in the output of the annotation tool, other information available in the corpus could be extracted and counted, such as the lemma of the context word, its dependency relation (or distance) to the target and its BOW distance to the target. For example, Tables 4.5 and 4.6 list the most frequent dependency paths, lemmas and window sizes across the official cues of *heilzaam* 'healthy/beneficial' for each of its senses. As we will see again in Section 6.2.1, this lemma is characterised by frequent nouns modified by the target, namely *werking* 'effect', *effect* and *invloed* 'influence', which are ambiguous in terms of the senses of *heilzaam*: in a sentence such as *de heilzame werking van look* 'the healing power of garlic', *garlic* is a better cue in the 'health/beneficial' distinction than *werking* 'effect, power'. Nonetheless, annotators did select these context words as cues for both senses, not realising that they were not distinctive of one or the other sense. The pattern fulfilled by *garlic* in this example was indeed captured by some cues, as shown in the third line of Table 4.5, but it is much less frequent.

Table 4.5: Four most frequent dependency paths among the cues of *heilzaam*, with counts per sense. `NA` indicates that the cue is not in the sentence of the target. In the path, `CW` stands for the cue and `T` stands for the target: the head is at the left of → and its dependents are to the right, preceded by the name of the dependency relation.

path	examples	beneficial	healthy
CW → mod:T	**heilzame** *werking* 'healing *power*'	60	41
NA	Different sentence	13	32
werking → [mod:T,mod:van → obj1:CW]	de **heilzame** werking van *look* 'the healing power of *garlic*'	8	14
ben → [predc:T,mod:voor → obj1:CW]	look is **heilzaam** voor de *gezondheid* 'garlic is beneficial for the *health*'	7	1

Table 4.6: Six most frequent lemmas and window spans among the cues of *heilzaam*, with counts per sense.

CW	healthy	beneficial	BOW	healthy	beneficial
werking/noun	20	12	1	74	48
effect/noun	5	9	4	38	28
gezondheid/noun	5	0	3	27	23
lichamelijk/adj	4	0	2	18	22
medisch/adj	4	0	5	21	20
economie/noun	0	4	6	21	14

4.3 Summary

In this chapter we looked at the dataset used to test and explore the workflow and the visualization tools. The selection of lemmas was described along with the semantic phenomena they would allow us to test. Afterwards, the annotation procedure was delineated, from the extraction of concordances to the assignment of senses, confidence values and cues.

As was mentioned before, for each of the lemmas, 200-212 models were generated following the workflow described in Chapter 2. The cues selected by the annotators informed some of the decisions involved in the parameter settings. The sense annotation was applied to assess how well the models performed at disambiguation: initially, we did not try to match senses to clustering solutions, but looked for a spatial configuration that might hide more subtle relationships. As a few examples in Chapter 3 have shown, this is much more straightforward in some lemmas than in others.

The range of semantic phenomena was meant to provide different possible aspects of meaning that distributional models might be able to capture. From a lexicological point of view, "similarity of distribution correlates with similarity of meaning" is not enough. What is similarity of meaning?[9] Does this mean that more granular distinctions, such as senses within homonyms, will be more difficult to capture than coarser distinctions, i.e. the homonyms themselves? Are metonymy, metaphor and specialization modelled by the same parameter settings? Can they be discriminated, can we fine-tune models to capture one or the other? And what is the role of constructions: does argument structure interfere in the modelling of senses? These were the questions that the case studies presented here tried to address, and the following part of this dissertation will present the answers.

[9]Sahlgren restricts the notion of meaning, as it can be found in distributional models, to "the meanings that are in the text" (2008: 49) and distinguishes between models that capture paradigmatic and syntagmatic relationships (without any further distinctions). Even if we *could* be satisfied with such an answer, it only applies to type-level models.

Part II

The cloudspotter's handbook

Chapter 5

A cloud atlas

Clouds come in many shapes. Like the cotton-like masses of droplets we see in our skies, the clouds of word occurrences generated by token-level distributional models may take different forms, depending on their density, their size and their distinctiveness. "Meaning is use", "Differences in usage correlate with differences in meaning", "You shall know a word by the company it keeps"[1] and other such catchy slogans sound intuitively accurate, but they hide a wealth of complexity and variation. Like meaning, context is far from orderly, and a myriad of words with different characteristics interact to generate the variation we see in these clouds.

In this chapter, we will try to make sense of the nephological topology, i.e. the variety of shapes that these clouds may take. For this purpose, we will classify HDBSCAN clusters mapped to t-SNE representations in a way that can help us understand what we see when we see a cloud. The starting point is the shape that a researcher sees in the t-SNE plot, which will be visually likened to types of meteorological clouds and further described in technical terms.

In Section 5.1 we will discuss the rationale behind this particular classification and the tools used to operationalize these decisions. A more detailed description of each cloud type and their technical interpretations follows in Section 5.2, while Section 5.3 zooms back out to compare the characteristics of the different types. Finally, we summarize the chapter in Section 5.4.

5.1 Rationale of the classification

When we look at the t-SNE plot of a token-level model, we might see different kinds of shapes. For example, Figure 5.1 shows the t-SNE solution for the same parameter configuration in six different lemmas. Some of them have clear, neat islands that stand out against a large mass, while others look smooth and uniform. Even this uniformity might take rounder or more angular shapes, with bursts of density when three or four tokens get together. As we have mentioned before, a t-SNE solution that looks very uniform typically means

[1] Attributed to Ludwig Wittgenstein, Zellig Harris and J. R. Firth respectively, as discussed previously.

that the perplexity is too high, whereas too many small islands suggest that it is too low. However, the models never seem to look better in the other perplexity values we have explored[2].

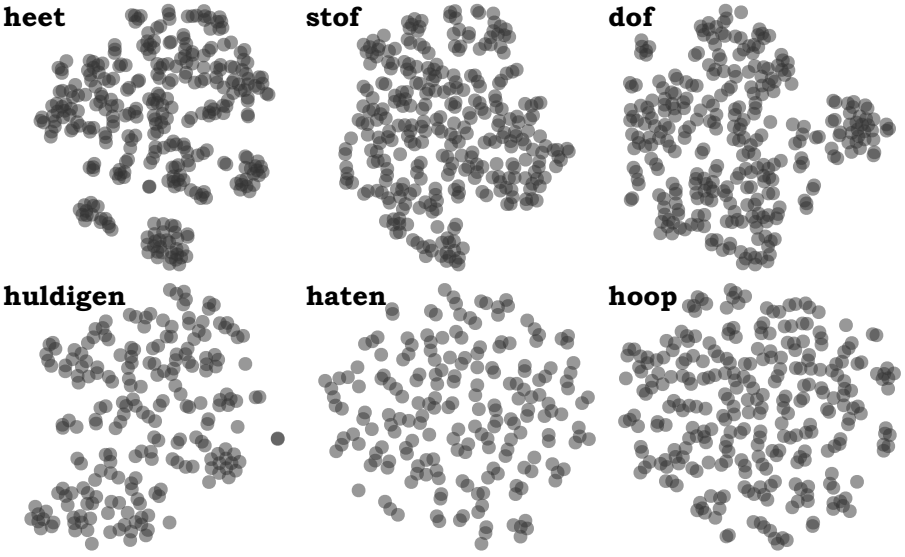

Figure 5.1: Uncoloured t-SNE representations of the same parameter settings (bound5lex-PPMIselection-FOCall) across six different lemmas.

Mapping an HDBSCAN clustering solution with $minPts = 8$, like we do in Figure 5.2 for the same models shown in Figure 5.1, has proved to be a decent system for identifying the structure we see in these clouds. Clusters tend to match the tighter islands we see, and to highlight dense areas that might be too subtle for our eyes. In some cases, the clustering solution and the visualization do not agree, e.g. clusters are spread around or overlap. This can be taken as a sign of uncertainty, as an indication that the group of tokens involved is much harder to describe and model that others in which both algorithms do agree.

At the stage of the distance matrix, we can establish, for each of our tokens, its similarity to any other token in the model. These similarities are independent from each other: until we do not transform them, they do not even need to respect the triangle inequality[3]. In contrast, both clustering and visualization add a layer of processing meant to find patterns of similar tokens that are different from the other tokens. The relationships between different pairs of tokens are not independent any more: nearest neighbours are the nearest because other tokens are farther away. Sometimes these patterns are easy to find, which leads to very nice, interpretable clouds, like the top plots in Figure

[2]This can be checked in Level 2 of the visualization: https://qlvl.github.io/NephoVis/.

[3]The triangle inequality refers to a property of metric spaces, according to which the distance between point A and point B cannot be larger than the sum of the distances between A and C and between B and C.

5.2; sometimes they are very hard to find, resulting in lots of noise and/or less defined clouds, like in the last two plots.

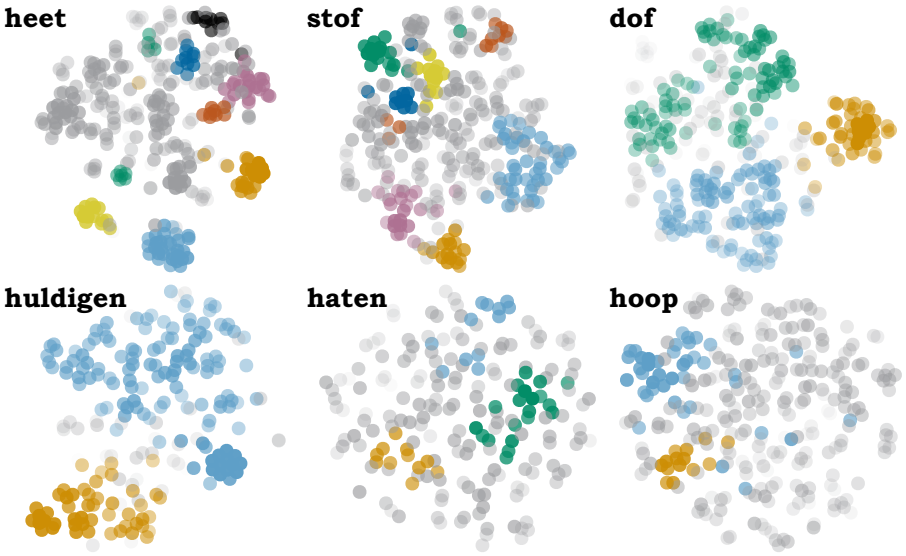

Figure 5.2: T-SNE representations of the same parameter settings (bound5lex-PPMIselection-FOCall) across six different lemmas, coloured coded with HDBSCAN clustering. Some of the *heet* clusters are gray because there are more clusters than colours we can clearly distinguish.

In this chapter we will look at a classification of the possible shapes, what we know about their genesis and how we can interpret them. The term *cloud* will refer to an HDBSCAN cluster or its noise: each of the coloured patches in the plots of Figure 5.2. The model itself, like a picture of the sky, might present multiple clouds of different types.

As we will see in Chapter 6 as well, the factors that interact to produce a group of similar tokens include the frequency of the context words, whether they co-occur within the sample and their type-level similarity. Clusters dominated by one context word may look similar to clusters dominated by a group of similar context words, and yet have different semantic interpretations. Along the way, we should keep in mind that the patterns observed here are tendencies, rather than rules: they are are a first map around an unknown land that still calls for more adventurous explorers.

The clouds have been classified into five main categories and an additional, orthogonal feature. The classification is based on a combination of t-SNE visualization (perplexity 30) and HDBSCAN ($minPts = 8$) and it would probably be different if other visualization techniques or clustering algorithms are used.

The main categories, which will be described in more detailed in Section 5.2, are, in descending degree of clarity:

- Cumulus: the most defined clusters, revealing strong patterns that t-SNE and HDBSCAN agree on;
- Stratocumulus: a slightly looser definition of still decent clouds;
- Cirrus: the weakest, smallest, less defined clouds, resulting from weaker patterns that might not be immediately evident without colour coding;
- Cumulonimbus: massive clouds;
- Cirrostratus: the HDBSCAN noise.

The inspiration for the names of the types of clouds is visual: the shapes that we would find when mapping the HDBSCAN clusters to the t-SNE solution resemble the shapes of different types of meteorological clouds. Admittedly, for those who are familiar with meteorological types of clouds, this is not necessarily the most salient feature. Altitude, temperature and composition, instead, are more relevant in categorizing metereological clouds. As we will see in Section 5.3, it could be possible to map the ε (epsilon) values to the altitudes of the clouds, but that might already take the metaphor too far.

Technical criteria were defined in order to automatically categorize a large number of clusters. They are the result of both theoretical reasoning and trial and error, so that the final classification matches the intuitions derived from visual inspection. In other words: this classification should help us understand what we are looking at based on the shapes we identify, but technical, objective criteria were designed that approximate these intuitions for a larger scale analysis. These criteria make use of (i) the noise category from HDBSCAN, (ii) the relative size of the cluster, (iii) separability indices, (iv) cosine distances between the tokens and (v) ε values.

Criteria (i) and (ii) are straightforward. Criterion (iii) refers to two measures developed within the `semvar` package (Speelman & Heylen 2014, 2017), `kNN` and `SIL`: they assess how well the items are clustered based on a distance matrix. In this case, we are looking for the match between the HDBSCAN clusters, which take the role of classes, and the euclidean distances within the t-SNE plot. Let's see how they work.

The first measure, `kNN`, is a separability index developed by Speelman & Heylen (2014) based on the proportion of "same class items" among the k nearest neighbours of an item. It answers the following question: looking at the HDBSCAN clusters mapped to the t-SNE plot: how pure are the clusters? Do they form tight groups of the same colour, or do they overlap (maybe with noise tokens)? Recall that this has no bearing on the semantic composition of the cluster: instead, it refers to the visual homogeneity of the cluster as mapped to the plot.

For our purposes, it makes sense to set k to 8, the minimum number of tokens that a cluster should have based on the current HDBSCAN parameters. As a result, for each token x of a cluster C, if the 8 tokens closest to x in the t-SNE plot belong to the same HDBSCAN cluster C, then $\texttt{kNN} = 1$, and if none of them do, then $\texttt{kNN} = 0$, regardless of what other class(es) the other items belong to. When the proportions are mixed, the ranking of the neighbours plays a role: if the tokens closest to x belong to C, `kNN` will be higher; if instead they belong to another class, `kNN` will be lower. The `kNN` value of the cloud itself (C) is the mean of the `kNN` assigned to each of its members. A high `kNN` means that there

are only a few instances of a different class mixed in among the tokens of the cloud: in other words, the cloud is quite compact and pure. The problem with `kNN` is that it is biased in favour of large clouds. The larger the cloud is, the higher the proportion of tokens that is entirely surrounded by items of the same cluster. However, clusters with the same `kNN` and different sizes have different shapes. In order to counteract this bias, we include a `SIL` threshold. `SIL`, or silhouette, is a popular measure of cluster quality that takes into account the distances between the members of a cluster and to the members outside that cluster (Rousseeuw 1987). When the tokens inside a cloud are much closer to each other than to tokens outside the cloud, `SIL` is highest, with an upper bound of 1. If the cloud is very spread out and/or other clouds are very close by, e.g. because they overlap, `SIL` will go down. Thus, a combination of high `kNN` and high `SIL` results in more compact, homogeneous, isolated clouds.

Criteria (iv) and (v) are the distances between the tokens belonging to the same cluster and the ε values respectively. The former refer to the original cosine distances between the tokens of the same cluster: the lower they are, the more similar the tokens are to each other. These may be different from the euclidean distances based on the t-SNE plot. Finally, ε values are extracted from the HDBSCAN clustering and were explained in Chapter 2.2.4. The lower the ε, the denser the area of the token, i.e. the smaller the area covered by its nearest neighbours. Noise tokens have typically the highest ε values: they are very disperse, and therefore the radius required to find 8 near neighbours is larger. The members of a cluster might have a variety of ε values: the lower the ε, the closer it is to the core, i.e. the denser area of the cluster. To be clear, I am not making any claims about the technical or semantic interpretation of ε right now. A brief discussion on this is given in Chapter 6. Instead, the utility of these values lies in their straightforward mapping to the visual effects of the plot. If the ε values of a clustered token are close to those of noise tokens, the cluster is, in a way, submerged in noise: HDBSCAN is finding patterns that t-SNE does not. On the contrary, if the ε values are much lower than for noise tokens, the cloud stands out.

The five criteria are combined in the following algorithm to classify the different clusters.

1. The noise is categorised as a Cirrostratus cloud.
2. The clusters that cover at least 50% of the modelled tokens (including noise) are Cumulonimbus clouds.
3. The clearest, roundest, densest clusters are Cumulus clouds. They must at least have a `kNN` ≥ 0.75, `SIL` ≥ 0.5 and mean cosine distance ≤ 0.5. In addition, less than 10% of the tokens in the cluster may have a higher ε than the lowest noise ε, or the noise in the model must cover less than 10% of the tokens.
4. The smallest clusters, i.e covering less than 10% of the modelled tokens, if 75% of the model is noise or `kNN` < 0.7, are Cirrus clouds.
5. The most decent of the remaining clusters are Stratocumulus clouds. They must have `kNN` ≥ 0.7, `SIL` ≥ 0.5 or mean distance ≤ 0.2. In addition, either more than half of the tokens have lower ε than the noise tokens or no more than 10% of the modelled tokens are noise.

Table 5.1: Number of clouds of each type per medoid or model in general; in parenthesis, the number of Hail clouds is specified.

Cloud type	Clouds in Medoids	All clouds
Cumulus	267 (25)	6899 (459)
Stratocumulus	412 (34)	8777 (692)
Cirrus	342 (2)	9477 (32)
Cumulonimbus	42 (15)	1025 (221)
Cirrostratus	254 (1)	6453 (3)

6. The remaining clusters are Cirrus clouds.

In addition, the category of Hail groups the clouds with at least 8 identical tokens; these can belong to any of the other classes.

Table 5.1 shows the number of clouds, either in medoid models or across all models, belonging to each of the categories. By definition, almost all models have a Cirrostratus cloud, i.e. noise tokens, and no more than one Cumulonimbus cloud, i.e. massive cloud. The rest of the clouds may occur more than once in the same model. The number of clouds that also belongs to the Hail category is given in parentheses.

5.2 Types of clouds

In this section, the different cloud shapes will be described in some detail. Their general look on a plot will be compared to pictures of meteorological clouds and I will offer a technical interpretation for them.

Before going into the descriptions, an explanation of one of the measures that takes part in the technical interpretation is in order: the F-score. Clouds can be represented by the set of context words co-occurring with the tokens that compose it. The relationship between each context word *cw* and the cluster may be described in terms of precision and recall, already mentioned in Section 3.5: **precision** indicates the proportion of tokens co-occurring with *cw* that also belong to the cluster, while **recall** indicates the proportion of tokens within the cluster that co-occur with *cw*. For example, if all the tokens in a cluster co-occur with the definite determiner *de*, *de* has a recall of 1 for that cluster; but in all likelihood, these tokens only constitute around 40% of the tokens co-occurring with *de* across the sample, resulting in a precision of 0.4. Both values can be summarized in an F-score, which is defined as the (weighted) harmonic mean of precision and recall. In this case, the unweighted F, that is, where precision and recall are deemed equally important, equals 0.57. The higher the F, the better the representativeness of the context word in relation to the cluster: an F of 1 indicates that all the tokens co-occurring with that word belong to that cluster, and all the tokens in that cluster co-occur with that word, while an F of 0 indicates the absolute lack of overlap between the domain of the context word and the clouds. When a context word has a high

F in relation to a cluster, that cluster is *dominated* by the context word. This is a handy term that will come up frequently as I describe types of clouds, and especially in Chapter 6. In general, only context words that co-occur with at least two tokens within a cluster are considered, to avoid inflating the value of *hapax legomena*.

5.2.1 Cumulus clouds

In meteorological terms, Cumulus clouds look puffy: they are our prototypical and ideal images of clouds. As token-level clouds, they also correspond to our ideal images of clusters: mostly roundish, visually salient because of their density and isolation. We would be able to find them even without colour-coding: both t-SNE and HDBSCAN agree that those tokens belong together. In Figure 5.3, the four rightmost clusters, in green, light blue, yellow and blue, are Cumulus; the rest are Stratocumulus clouds.

Figure 5.3: Example of Cumulus cloud: inspiration on the left, plot example on the right (nobound10lex-PPMIweight-FOCall of *dof*). Picture by Glg, edited by User:drini - photo taken by Glg, CC BY-SA 2.0 de, https://commons.wikimedia.org/w/index.php?curid=3443830.

Cumulus clouds are defined by a number of different measures with strict values, after excluding Cirrostratus (noise) and Cumulonimbus (massive clouds). First, the clusters need to have both `kNN` ≥ 0.75 and `SIL` ≥ 0.5, as well as a mean pairwise cosine distance between the tokens of 0.5 or lower. The combination of these three strict thresholds ensures quite pure, compact, isolated clusters: they don't visually overlap with other clusters or noise. The final requirement makes sure that the cloud stands out against the noise. One of the ways it can achieve this is by having an ε lower than the minimum noise ε in at least 90% of the tokens: at least 9 out of 10 tokens stand out. However, in models without any noise or with very little, noise ε values might be particularly high, so this threshold is not applied in models with less than 10% noise.

Most of these clouds are characterized by one context word with high precision and recall for the cluster. In fact, 75% of these clouds have a context

word with an F of 0.72 or higher, while in 75% of the rest of the clouds the highest F is lower than that. These top context words also tend to have high PMI, but some may even have negative PMI.

The lemmas with the highest proportion of Cumulus clouds are *heffen* 'to levy/to lift', *hachelijk* 'dangerous/critical', *schaal* 'scale/dish', *gemeen* 'common/mean…' and *stof* 'substance/dust…'. They are all cases with strong collocational patterns of the kind discussed in Section 6.2. Lemmas that repel Cumulus clouds, on the other hand, such as *haten* 'to hate', *geestig* 'witty', *gekleurd* 'coloured' and *hoekig* 'angular', lack such collocational patterns and instead form more uniform, fuzzy pictures.

5.2.2 Stratocumulus clouds

In meteorological terms, Stratus clouds are flat or smooth clouds: Stratocumulus clouds are then a flatter, less compact version of the Cumulus clouds discussed above. In Figure 5.4, all three clouds are Stratocumulus: from the large, disperse light blue cloud, to the more stretched orange one and the more compact green cloud that lost three points in the bottom right.

Figure 5.4: Example of Stratocumulus cloud: inspiration on the left, plot example on the right (bound5all-PPMIno-FOCall of *heffen*). Picture by Joydeep - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=20357040.

The definition of Stratocumulus clouds takes a number of different measures and applies less strict thresholds than for Cumulus clouds. First the Cirrostratus, Cumulonimbus and Cumulus must classified, and the smallest clouds, either in noisy models or without high `kNN`, must be reserved for Cirrus. On the remaining clouds we apply two filters. First, they must either have `kNN` ≥ 0.7, `SIL` ≥ 0.5 or mean pairwise cosine distance ≤ 0.2. Second, either more than half the tokens have an ε value below the minimum noise ε value or the percentage of noise tokens in the model is lower than 10%.

Stratocumulus clouds are generally large: while 75% of either Cumulus or Cirrus have 28 tokens or fewer, half the Stratocumulus have 26 or more. However, in comparison to Cirrus they tend to have lower type-token ratio of

context words[4] and higher F values of their representative context words. In addition, the mean cosine distance between the tokens tend to be comparable to that in Cirrus clouds, in spite of the difference in size: in other words, they are larger but more compact and more clearly defined.

While lemmas that prefer Cumulus clouds tend to avoid Cirrus clouds and vice versa, the relationship with Stratocumulus is not so straightforward. A preference for Cumulus tends to go hand in hand with a preference for Stratocumulus, as in the case of *heffen* 'to levy/to lift', but that is not necessarily the case. Both *gemeen* 'common/mean…' and *stof* 'substance/dust…' prefer Cumulus against either Cirrus or Stratocumulus, and *haten* 'to hate', which prefers Cirrus to Cumulus, does have a slight preference for Stratocumulus too. One lemma that prefers Stratocumulus over anything else is *heilzaam* 'healthy/beneficial', which is described in Section 6.2.1: even though its clusters tend to be dominated by clear collocates of the target, they are semantically heterogeneous.

5.2.3 Cirrus clouds

From a meteorological perspective, Cirrus clouds are high up and wispy. In these plots, the description translate to typically small, disperse clouds that we might not be able to isolate without the help of HDBSCAN. In Figure 5.5, both clouds belong to this category.

Figure 5.5: Example of Cirrus cloud: inspiration on the left, plot example on the right (bound5all-PPMIselection-FOCall of *herstructureren*). Picture by Dmitry Makeev - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=85153684.

Cirrus clouds are defined as small clouds in noisy models or with a low `kNN`, i.e. substantial overlap between the cloud and other clusters or noise tokens, as well as the remainder of the clouds after defining the other four categories. They are generally small, like Cumulus clouds: in a few cases they cover more than 100 points, and they would be considered Stratocumulus if their `SIL` was higher and either their `kNN` or the percentage of tokens below noise was higher

[4]Either counting all context words in the cluster or just those that are enough to cover the tokens.

too. In spite of their size, they have a high type-token ratio of context words and these context words have low F, even compared to larger Stratocumulus clouds: in other words, they tend not to be represented by single powerful collocates, and instead their tokens co-occur with many different, infrequent words.

The weakness of their patterns should be seen as a tendency, rather than a law. They are more likely than Cumulus clouds to be semantically heterogeneous and hard to interpret, but it is not necessarily the case. In some lemmas with tendency to a more uniform internal structure, Cirrus clouds may group the few patterns that emerge at all. Lemmas that prefer Cirrus clouds, such as *geestig* 'witty', *gekleurd* 'coloured', *hoekig* 'angular' and *haten* 'to hate', are precisely characterized by uniform-looking plots, low frequency collocates and weak patterns overall.

5.2.4 Cumulonimbus clouds

In the physical world, Cumulonimbus clouds are puffy (as indicated by the prefix *Cumulo-*) and bring rain and storm (*nimbus*). They are massive, towering clouds that may lie as low as Cumulus clouds and reach as high as Cirrus clouds. In our models, the Cumulonimbus category (the largest cluster in Figure 5.6) is the least frequent, but when it does occur, it dominates the picture.

Cumulonimbus clouds are minimally defined as clouds that cover at least 50% of the modelled tokens, including those discarded as noise. In practice, half of them cover at least 58.7% of the model or more, reaching as much as 95.7%. Next to them, we typically have one more cluster (in 85.6% of the cases); occasionally we may have two (11.1%) or even up to 5. The smaller cluster next to the massive Cumulonimbus tends to be a Cumulus, but all combinations are attested.

Figure 5.6: Example of Cumulonimbus cloud: inspiration on the left, plot example on the right (bound10all-PPMIweight-FOCall of *stof*). Picture by fir0002flagstaffotos [at] gmail.comCanon 20D + Canon 17-40mm f/4 L, GFDL 1.2, https://commons.wikimedia.org/w/index.php?curid=887553.

The most typical situation in which we encounter a Cumulonimbus cloud is when a small group of tokens is very tight, but very different from everything else, and the rest of the tokens are not distinctive enough to form different

clusters. Most of these tokens are then grouped in this large, normally disperse Cumulonimbus cloud, which may seem to have inner structure captured by t-SNE but not by HDBSCAN. The small group of tokens may be brought together by a set of similar context words (see Section 6.4), but most typically they represent an idiomatic expression.

In fact, the lemmas with a strong preference for this format, with Cumulonimbus in more than a third of their models, have a very clear idiomatic expression responsible for the small Cumulus clouds, so that the differences among the rest of the tokens are smoothed. These are *gemeen* 'common/mean…', *stof* 'substance/dust…' and *schaal* 'scale/dish'. In contrast, lemmas that barely have any Cumulonimbus clouds (in less than 5% of the models), such as *herroepen* 'to recant/to void', *hoekig* 'angular', *diskwalificeren* 'to disqualify' and *horde* 'horde/hurdle', lack such a strong pattern and have groups with similar frequencies and mutual differences instead.

In the case of *gemeen* 'common/mean…', the tight cloud represents the expression *grootste gemene deler* 'greatest common divisor': both *groot* 'big, great' and *deler* 'divisor' co-occur with a large number of tokens but are, at the type-level, different from each other and to everything else. As a result, the token-level vectors of the *grootste gemene deler* 'greatest common divisor' tokens will be very similar to other tokens instantiating the same expression, and very different from everything else. Similarly, the pattern most frequently tied to this phenomenon in the case of *stof* 'substance/dust…' is *stof doen opwaaien* 'lit. to stir up dust', an idiomatic expression referring to controversial actions and situations. *Schaal* 'scale/dish', on the other hand, has two main idiomatic contexts that generate Cumulonimbus clouds, discussed in Section 6.2.2.

The rest of the tokens, i.e. the Cumulonimbus cloud itself, is not defined by either a strong dominating context word or group of similar context words, but instead is defined against this stronger, small cloud. Cumulonimbus clouds are not huge clouds of similar tokens, but a mass of tokens that is not structured enough in opposition to the distinctive small cloud that is next to it. It may have dense areas inside of it, but they are not semantically linked to each other. The reason they are a cluster is not because the tokens are similar to each other, as much as because the tokens in the small partner are very coherent and different from everything else. The mean distance between tokens in a Cumulonimbus cloud is typically very large, sometimes as large as within Cirrostratus (noise), and significantly larger than within other kinds of clouds — although the few examples of Cirrus and Stratocumulus co-occurring with Cumulonimbus also have relatively large mean distances. For a discussion on the semantic interpretation of these clouds, see Section 6.5.

5.2.5 Cirrostratus clouds

In meteorological terms, Cirrostratus clouds are high (*Cirro-*), flat and smooth (*-stratus*) clouds. For our purposes, they just indicate the noise tokens. They lie in the background of (almost) all our clouds and constitute 100% of two of the medoids. Considering the entirety of the models, 146 (2.3%) of them are

fully Cirrostratus clouds (Figure 5.7).

Figure 5.7: Example of Cirrostratus cloud: inspiration on the left, model with 100% noise on the right (nobound10lex-PPMIno-FOCall of *hoopvol*). CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=100381.

It might be interesting to cluster the subset of tokens that make up these clouds, at least for some lemmas, but that is not pursued in these studies. It would require a deeper investigation of how HDBSCAN works with these models, why tokens are sometimes not clustered and how it interacts with parameters like $minPts$. I will not try to semantically interpret these clouds, but they are always present and affect how other clouds are defined.

5.2.6 Hail

The final, orthogonal category can apply to any cloud, and often describes a section of it rather than the full cloud. It responds to a special criterion, to highlight the occasional phenomenon of superdense clusters. In Figure 5.8, three of the clouds (light blue, yellow and red) are Cumulus, while the rest are Stratocumulus; all of them but the yellow and green clouds present Hail, that is, extremely tight, dense circles of identical tokens. These are clouds with at least 8 identical tokens, defined as having a cosine distance lower than 10^{-6}.

As we can see in the blue cloud, one cluster may have more than one piece of Hail, as is the case in some Cumulonimbus clouds. In fact, in relative numbers, the cloud type with a higher tendency to generate Hail is the Cumulonimbus (in 21.56% of the cases), which is very fitting for a cloud that brings storms. Overall, 5% of the clouds, in 924 different models, have these characteristics.

These conditions are prompted by a low number of context words per token and a low type-token ratio (TTR) of these context words (see Figure 5.9). TTR is a measure of complexity computed as the number of different context words, i.e. types, divided by the total number of occurrences, i.e tokens. A TTR of 1 indicates that all words are only used once, while a lower TTR results from different words occurring multiple times. In this case, the higher the TTR, the richer the variety of context words captured by the model for the tokens in the cluster. Hail only covers a minority of the clouds, but it is clear that both the

Figure 5.8: Example of cloud with hail: inspiration on the left, plot example on the right (REL1-PPMIselection-FOCall of *heet*). Picture by Tiia Monto, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=88743807

TTR and the number of context words per token play a role, with lower values for the Hail clouds. Hail tends to emerge in very restrictive models where many tokens can be grouped together because they have identical vectors: they shared the few words that survived the thresholds. They often reveal the strongest context words, i.e. those that also dominate in other clouds. But as we will see in Section 6.2, the dominating context word is not always indicative of a sense. Moreover, a larger variation in the context can give us a richer, more nuanced picture of the distributional behaviour of the target word.

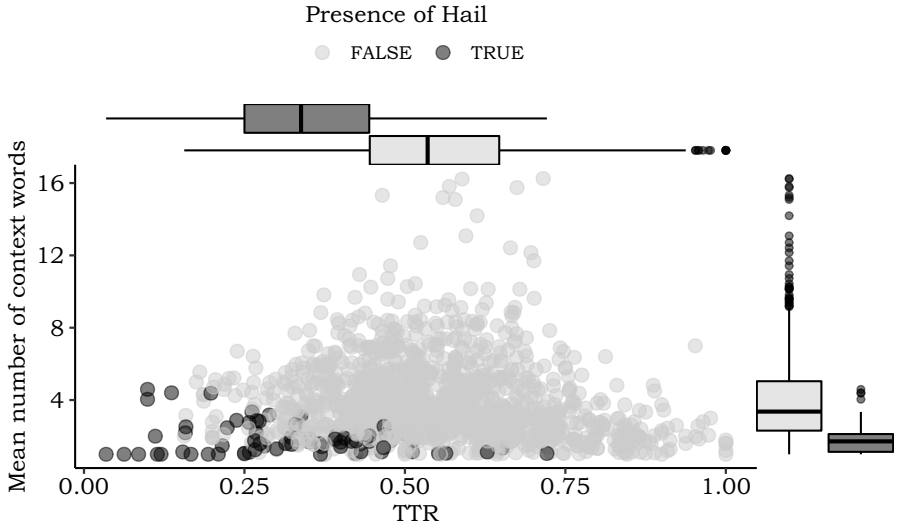

Figure 5.9: Mapping between the type-token ratio of the context words and the mean number of context words per token in a cluster of a medoid, by whether the cloud has Hail.

We might be tempted to consider these clouds idiomatic expressions: they match, visually, what we think a representation of idiomatic expressions would be like. Instead, they match groups of context words that occur very frequently in a very short distance (either in terms of bag-of-words or dependency relations) to the target. It tells us something about bigrams: about how often *niet* 'not' occurs close to *harden* 'to tolerate'; *van* 'of' to *staal* 'steel/sample', *op* 'on' to *spot* 'spotlight', or *hang_ijzer* 'iron' to *heet* 'hot'. At the same time, all the *harden* 'to tolerate' tokens co-occurring with *niet* 'not' are brought together in one pattern that disregards any other possible co-occurrence, any possible internal variation.

5.3 Patterns across types of clouds

Beyond the features used as formal criteria to define the types of clouds, we can find patterns across other relevant features. Most of these features are technical properties that can be extracted automatically from our dataset: the representativeness of context words F, their PMI with the target lemma, the type-token ratio of context words co-occurring with the tokens in a cluster (TTR), and their ε values. In addition, we will take advantage of the semantic annotation and look at the entropy of the clouds, i.e. how homogeneous they are in terms of dictionary senses. The figures in this section represent clusters in the medoids, because including all the models results in a more cluttered version of the same patterns.

First, we will look at the properties of the most representative context word in each cluster, defined as the context word with a minimum frequency of 2 within the cluster and the highest F-score (explained in Section 5.2). Figure 5.10 plots the context words' F on the horizontal axis and, on the vertical axis, their PMI with the corresponding target lemma, based on a symmetric window of 4 words to either side. The types of clouds are mapped to the colour of the points and the fill of the marginal boxplots. Therefore, a bright purple spot at $x = 0.941$ and $y = 9.89$ represents a Cumulus cloud whose most representative context word, e.g. *deler* 'divisor', has an F of 0.941, i.e. it is a very good cue, and a PMI with the lemma of its model, e.g. *gemeen*, equal to 9.8, which is very high.

We can see that Cirrostratus clouds (noise) tend to have low F context words, and that these tend to have very low PMI values with the targets. Cumulus clouds, on the other side, have the highest F, i.e. they tend to be dominated by one context word, and these context words tend to have high PMI. Cirrus and Cumulonimbus clouds have lower values than Cumulus clouds across both dimensions, while Stratocumulus spans in between. Moreover, the correlation between F and PMI is moderate to weak, with a higher value among Cumulus clouds (Spearman's $\rho = 0.55$, p-value < 0.001) and much lower and/or less significant for Cumulonimbus and Cirrostratus ($\rho = 0.05$, p-value ≈ 0.085 and $\rho =$ -0.15, p-value < 0.001, respectively), with minimal changes for different `PPMI` settings.

Second, we will look at the relationship between the highest F-score and the type-token ratio (TTR, described in Section 5.2.6). They are mapped to

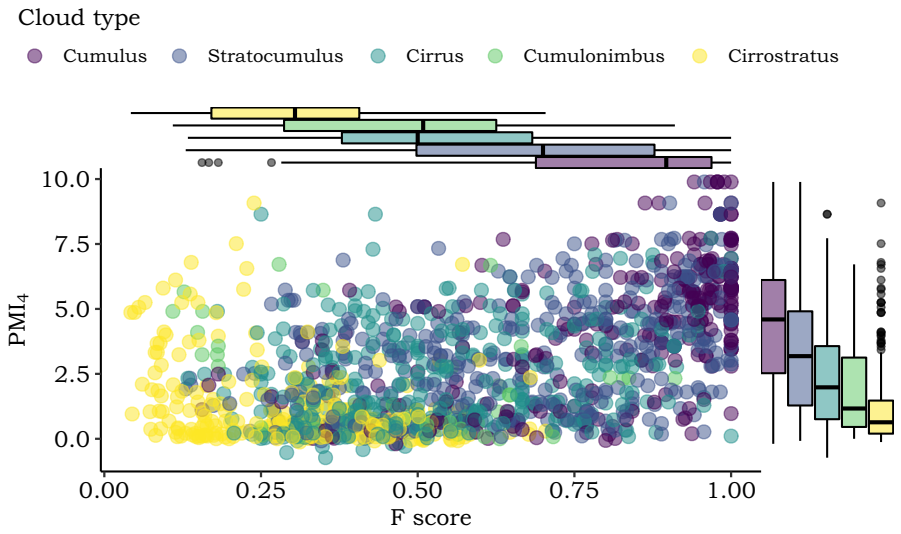

Figure 5.10: Mapping between the highest F between the clouds of the medoids and a context word and that context word's PMI with the target, coloured by cloud type.

the horizontal and vertical axes respectively in Figure 5.11. As we can see, both Cirrus and Cirrostratus clouds tend to have a higher TTR than the rest; indeed, they also tend not to have Hail (see Table 5.1). In general, the more different context words we find in the cluster — the higher the TTR —, the lower is the representativity of the strongest context word — the lower the F —, but there is a really wide range of variation, and each type of cloud has a different profile. Cirrostratus clouds have higher TTR and lower F, while the opposite defines Cumulus clouds; Stratocumulus and Cumulonimbus clouds have similar TTR to Cumulus but lower or much lower F, and the TTR in Cirrus clouds is comparable to Cirrostratus, with a much higher F-score. In short, both the variety of context words co-occurring with the tokens of a cluster and the F-score of the most relevant of them play a role in the shape that the clouds take. This relationship notwithstanding, we should note that other factors also intervene, both in the constitution of the clusters and their semantic interpretation, such as the representativeness of other context words and the type-level distances between them.

Third, we will look at the ε values across different cloud types. They were part of their definition, insofar the proportion of tokens with a lower ε than the lower noise ε is a criterion for Cumulus, Stratocumulus and Cirrus clouds. Nevertheless, it could be useful to summarize the resulting patterns. In that spirit, Figure 5.12 shows, for each of the clusters in the medoids, the minimum, mean and maximum ε within a cluster. Note that this gives us no insight on the relationship between the ε of the tokens in the cluster and the values in other

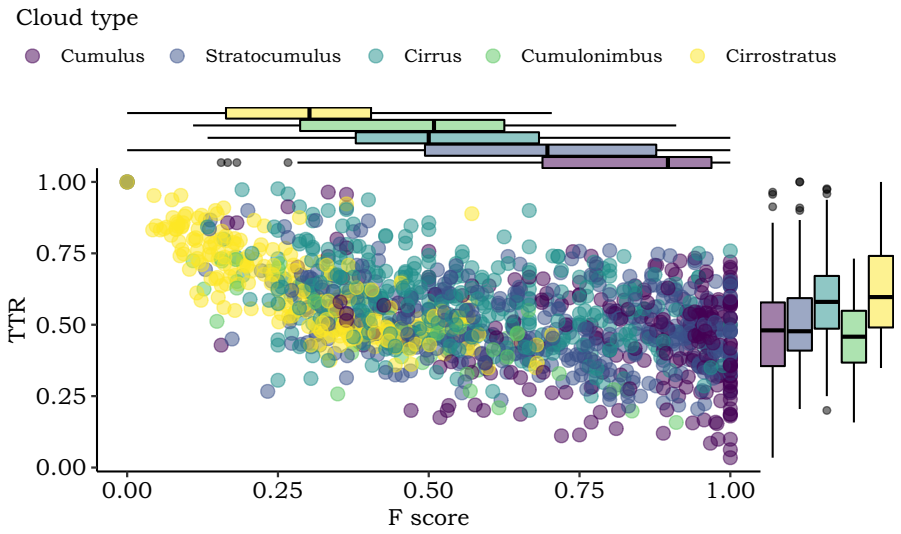

Figure 5.11: Mapping between the highest F of a context word to a cluster and the type-token ratio (TTR) of context words in the cluster, coloured by cloud type.

tokens, either clustered or noise. As we would expect, Cirrostratus clouds tend to have the highest ε: their tokens are disperse, far away from other tokens. Cumulus, Stratocumulus and Cirrus tend to have relatively similar values, although some Cumulus clouds are quite low and Cirrus clouds are quite flat and never very low. The differences between them are more likely to be found in their relationship with the noise ε. Cumulus clouds stand out as dense areas against a very disperse (i.e. high ε) background, whereas Cirrus clouds are just slightly denser than the rest of the clouds in their surroundings. Think of the uniform plots of *haten* and *hoop* in Figure 5.1 and the HDBSCAN interpretation in Figure 5.2: all the tokens look, roughly, equally disperse. Finally, Cumulonimbus clouds exhibit the widest range of all: their tokens vary from very dense areas, i.e. very low ε, to very disperse ones, i.e. high ε.

These characteristics can be roughly mapped to the altitude of the clouds from a meteorological perspective (see Figure 5.13). Figure 5.12 relies on a common everyday metaphor: high values above, low values below. But what counts as high and low here is rather arbitrary: low ε indicates a dense area in the plots, that is, that a token can find its 7 nearest neighbours in a very small radius, while a high ε indicates a sparse area, with large distances between a token and its nearest neighbours. At the same time, the metaphor of altitude is also coherent with the original goal of this mapping: the low values, or high densities, stand out to the viewer as if they were closer. Similarly, if we look at clouds on the sky from below, lower clouds are going to stand out: that is the case of Cumulus, Stratocumulus and, naturally, Cumulonimbus clouds.

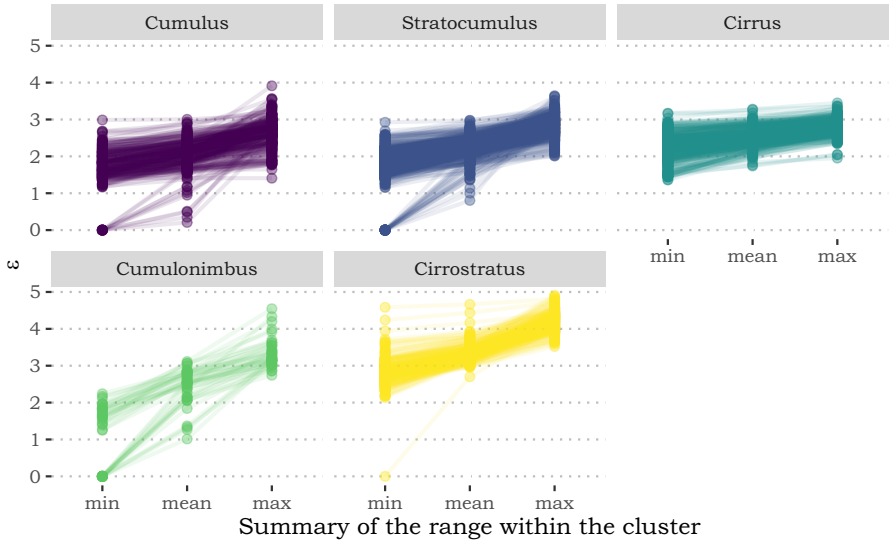

Figure 5.12: Range of ε values within clouds of different types. Lines join the points belonging to the same cluster.

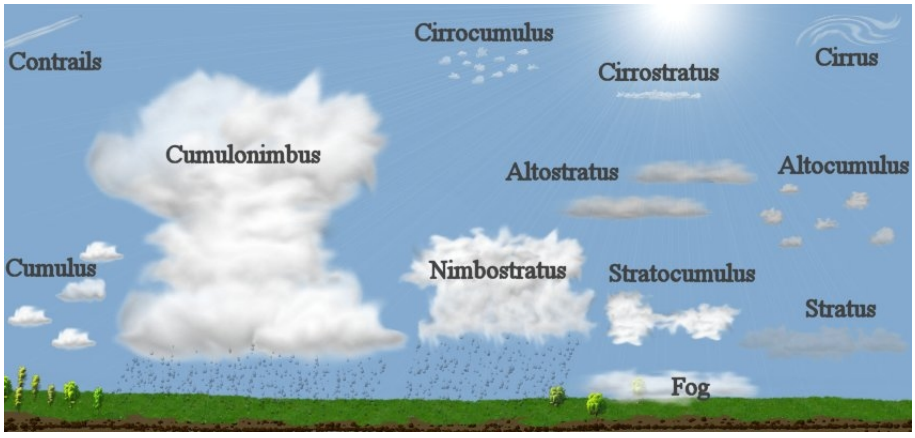

Figure 5.13: Graphical representation of meteorological clouds at different altitudes. By Christopher M. Klaus at w:en:Argonne National Laboratory - Own work by en:User:Klaus, Public Domain, https://commons.wikimedia.org/w/index.php?curid=2760873

Finally, Figure 5.14 shows the entropy[5] of clouds of different types in terms of the manually annotated senses, against the entropy across the whole model. Entropy is a measure of information, and in this case it works as follows: the higher the entropy, the more variation of senses and the more balanced their frequencies; the lower the entropy, the more one sense dominates. We would like the entropy of the cluster (the y-axis in Figure 5.14) to be as low as possible, that is, for the cluster to be as homogeneous as possible in terms of senses. At the same time, models with a higher initial entropy — due to the sense distribution of the lemmas they model — are more likely to have clusters with higher entropy.

On the one hand, the horizontal boxplots show that clouds of all types are equally likely to emerge in any model regardless of their sense distribution. This is consistent with the point I make in Chapter 6 that clouds do not model senses. On the other hand, the vertical boxplots show that the different cloud types do tend to exhibit different entropy values. Cumulus clouds are the most homogeneous, while the Cumulonimbus clouds tend to be as heterogeneous as Cirrostratus (noise) — they might even have higher entropy than their models as a whole. Cumulus clouds in particular, but sometimes also Stratocumulus and maybe Cirrus, may be completely homogeneous regardless of the sense composition of the model itself, but they can also have higher entropy. Stratocumulus clouds tend to have slightly lower entropy than Cirrus clouds, i.e. they tend to be more homogeneous, even though they also tend to be larger.

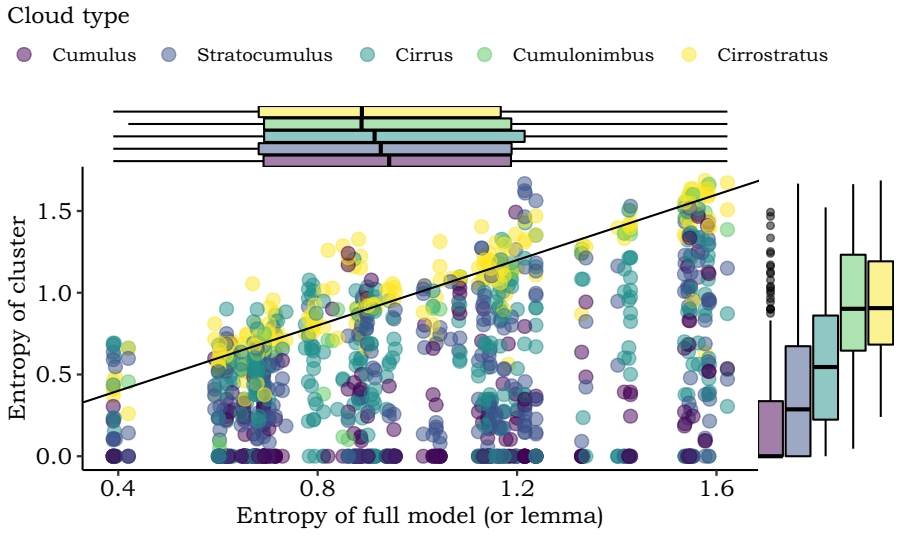

Figure 5.14: Mapping between the entropy in a medoid and in a cluster by cloud type.

[5]Computed with `entropy::entropy()` (Hausser & Strimmer 2021).

5.4 Summary

Token-level distributional models process a wealth of complex, messy information and try to return clear, interpretable patterns. These patterns take different forms: sometimes we have very clear, isolated, dense groups of similar tokens, like our ideal image of clouds in a clear sky; other times, a pattern is harder to find, and we barely catch a few clear wisps against an overcast sky. The clouds of one model are not independent from each other, and depending on the power of their leading context words they might merge into larger masses or split into smaller groups; powerful small Cumulus clouds may force everything else into a huge Cumulonimbus clouds, and the tension between its semantic fields might even create Hail.

In this chapter we have seen the variety of shapes that emerges from these distributional models, in particular in the interaction between the t-SNE visualization with perplexity 30 and the HDBSCAN clustering with $minPts = 8$. We have linked the visual results to the variety of context words and their representativeness, and we have found patterns in their semantic homogeneity. We know that all lemmas exhibit all types of clouds, but in different proportions, related to their tendency towards strong collocational patterns. In the next chapter, we will delve into the linguistic interpretation of these clouds, that is, their collocational properties and their relationship to the manually annotated dictionary senses.

Chapter 6

The language of clouds

In linguistic terms, clouds may provide us with different types of information, both at syntagmatic and paradigmatic level. At the syntagmatic level, they may illustrate cases of collocation, colligation, semantic preference or even tendencies towards the open-choice principle. The paradigmatic level, on the other hand, codes the relationship between the clusters and dictionary senses, from heterogeneous clusters to those that represent (proto)typical contexts of a sense.

Given a naive understanding of the correlation between context and meaning, we would mostly expect, from the paradigmatic perspective, clusters that equal senses: each cloud would cover all the occurrences of a dictionary sense and only the occurrences of that sense. However, even if we relax the requirements, expecting *mostly* homogeneous clusters covering *most* of the *clustered* tokens, this does not arise often. Instead, even homogeneous clusters only group typical contexts within a sense, which, at the syntagmatic level, tend to correspond to collocations. In any case, as we will see in this chapter, the full picture is more complex, and we can obtain much richer information than just lexical collocations representing typical contexts within a sense.

In this chapter, we will look into the types of syntagmatic and paradigmatic information that the clouds offer. Section 6.1 starts with an overview of the different levels in each dimension and mentions a few examples of their interaction in a contingency table. We then elaborate with more detailed examples of each in situation in sections 6.2 through 6.5, and round up with an overall summary in Section 6.6.

6.1 Types of information

The linguistic information obtainable from the clusters can be understood from the syntagmatic perspective as co-occurrence patterns of different kinds, and from the paradigmatic perspective in relation to dictionary senses. Both dimensions interlace, resulting in a number of specific phenomena that we may encounter. The relationship is summarized in Table 6.1; the syntagmatic or collocational dimension is represented by the columns and discussed in Section

6.1.1, and the paradigmatic or semantic dimension is represented by the rows and discussed in Section 6.1.2.

6.1.1 Collocational perspective

In order to interpret the different levels of information that a syntagmatic or collocational perspective may offer us, we can make use of some theoretical concepts from the foundations of Corpus Linguistics. Some of the terms were already coined by Firth (1957), but they were integrated in a framework for corpus analysis by Sinclair (1998: 124-125) and other publications. The framework includes, next to the node, i.e. our targets, four key components: one obligatory — semantic prosody, which will not be discussed here — and three more that will help us make sense of the observed output of the clouds: collocation, colligation and semantic preference.

In their simplest form, **collocations** are defined as the co-occurrence of two words within a certain span (Firth 1957: 13, Sinclair 1991: 170, 1998: 15, Stubbs 2009: 124). They might be further filtered by the statistical significance of their co-occurrence frequency or by their strength of attraction; such as PMI (see McEnery & Hardie 2012: 122-133 for a discussion). Even though a collocational relationship is asymmetric, that is, the co-occurrence with a more frequent word B may be more important for the less frequent word A than for B, the measures used to described it are most often symmetrical (Gries 2013). When it comes to the interpretation of clouds, this category takes a different form and is definitely asymmetric. Considering models built around a target term or node, frequent, distinct context words are bound to make the tokens that co-occur with them similar to each other and different from the rest: they will generate clusters. Such context words do tend to have a high PMI with the target, but, crucially, they stand out because they are a salient feature among the occurrences of the target, independently from how salient the target would be when modelling the collocate. Concretely, we are talking about clusters defined by one context word or a group of co-occurring context words with a high F-score in relation to the cluster: these context words can be interpreted as collocates of the target. Unlike in most collocational studies, where you study a list of words that co-occur (significantly) frequently with your target node, vector space models allow you to see whether these context words exclude each other or also co-occur within the context of the target. In fact, we might even find more complex collocational patterns, including multiple context words.

Whereas collocation is understood as a relationship between words (and, traditionally, as a relationship between word forms), **colligation** is defined as a relationship between a word and grammatical categories or syntactic patterns (Firth 1957: 14, Sinclair 1998: 15, Stubbs 2009: 124). In order to capture proper colligations as clusters, we would need models in which parts of speech or maybe dependency patterns are used as features, which is not the case in these studies. However, by rejecting a strict separation between syntax and lexis (for everything is semantics in Cognitive Linguistics), we can make a grammatically-oriented interpretation of collocations with function words, such as frequent prepositions or the passive auxiliary. Given this caveat, we will talk

about lexically instantiated colligations when we encounter clusters dominated by items that indicate a specific grammatical function.

Semantic preference is defined as the relationship between a word and semantically similar words (Sinclair 1998: 16, Stubbs 2009: 125, McEnery & Hardie 2012: 138-140). Within traditional collocational studies, this implies grouping collocates, that is, already frequently co-occurring items, based on semantic similarity, much as colligation can be the result of grouping collocates based on their grammatical categories. Compared to collocation, its identification requires more interpretation on the part of the researcher. In the interpretation of individual clusters, semantic preference appears in clusters that are not dominated by a single collocate or group of co-occurring collocates, but are instead defined by a group of infrequent context words with similar type-level vectors and for which we can give a semantic interpretation. (Cases of similar context words without a semantic interpretation are quite rare, and normally involve pronouns or adverbs.) This is a key contribution of token-level distributional models that may remain inaccessible in traditional collocational studies: next to powerful collocates that group virtually identical occurrences, we can identify patterns in which the context words are not the exact same but are similar enough to emulate a collocate.

The three notions described above assume identifiable patterns: occurrences that are similar enough to a substantial number of other occurrences, and different enough from other occurrences, to generate a cluster. Going back to Sinclair (1991)'s founding notions, we are assuming the domination of the idiom principle:

> …a language user has available to him or her a large number of semi-preconstructed phrases that constitute single choices, even though they might appear to be analysable into segments. (Sinclair 1991: 110)

The opposite situation would be given by the open-choice principle:

> At each point where a unit is completed (a word or a phrase or a clause), a large range of choice opens up and the only restraint is grammaticalness. (Sinclair 1991: 109)

The idiom principle and the open-choice principle are supposed to organise the lexicon and the production of utterances. But if, instead, they are understood as poles in the continuum of collocational behaviour, they can help us interpret the variety of shapes that we encounter within and across lemmas. Lemmas in which we tend to find identifiable clusters, with strong collocations, lexically instantiated colligations or sets with semantic preference, can be said to respond to the idiom principle. In contrast, lemmas that exhibit large proportions of noise tokens, and small, diffuse clusters (Cirrus clouds, mostly), can be said to approximate the open-choice principle. They don't necessarily lack structure, but whatever structure they have is less clear than for other lemmas, and harder to capture with these models. With this reasoning, next to the three categories described above, we include **near-open choice** as a

fourth category, meant to include the clouds that do not conform to either of the clearer formats.

6.1.2 Semantic perspective

In terms of the relationship between the HDBSCAN clusters and the manually annotated dictionary senses, we can initially distinguish between heterogeneous clusters, i.e. those that do not exhibit a clear preference for one sense, and homogeneous clusters. Secondly, the homogeneous clusters may cover all the (clustered) tokens of a given sense, or only part of it, i.e. a (proto)typical context of the sense. Additionally, said (proto)typical context may highlight a certain aspect or dimension of the meaning of the target, different from that highlighted by a different context.

As a result, the semantic dimension covers four different types of situations. The first one, i.e. heterogeneous clusters or clusters with multiple senses, would normally be interpreted as bad modelling, if we consider the senses a gold standard and the target of our models. It is also the most frequent interpretation of the near-open choice clouds. Nonetheless, they can also occur in other kinds of clouds, and as such illustrate the mismatch between contextual and semantic structure: clear contextual patterns do not imply dictionary senses. The second type of situation, i.e. clusters that perfectly match senses, is the ideal situation and what we would initially expect from distributional models. Instead, it is quite rare and often indicative of fixed expressions or very particular meanings. Rather than full senses, contextual patterns tend to represent (proto)typical contexts of a sense.

As it was already described in Section 1.2.2, the notion of prototypicality in Cognitive Semantics is related to the principle that categories need not be discrete and uniform and to its application to the semasiological structure of lemmas and their meanings (Geeraerts 1988, 1997). At the extensional level, which in this case covers the domains or contexts of application of our target item, categories may be defined by a varied set of overlapping features (i.e. context words) and have fuzzy boundaries and/or degrees of membership. The central or more prototypical members of this category exhibit more of these overlapping features; the fewer features co-occur with an item, the weaker its connection to the category. As they appear in the clouds, a sense may exhibit one typical context that is much more frequent and clear that the rest, or multiple typical contexts with similar frequencies. Unfortunately, neither t-SNE nor HDBSCAN provide a reliable mapping between quantitative properties and relative centrality *of* the clusters. In contrast, we can identify central cases within an HDBSCAN cluster based on their membership probability, which, as briefly mentioned before, is the normalized core distance within a cluster. Items with a higher membership probability lie in a denser area of an HDBSCAN cluster, and therefore have more items similar to it than the items in sparser areas. They do not necessarily occur in the euclidean centre in the t-SNE plot, but might form one or more dense cores closer towards an edge instead. In addition, we can distinguish between rather uniform clusters, in which all members have a similar weight, from more diverse clusters with dense cores

and sparse peripheries.

Extensional prototypicality works at multiple levels. We could identify (proto)typical instances/contexts of a lemma, of a particular sense, or of a dimension of a sense. In this last case, we run into an interaction with intensional prototypicality. On the one hand, we find multiple extensionally prototypical patterns, i.e. two or more groups of attestations that instantiate different patterns. On the other, each of these patterns correlates with a different semantic dimension or aspect, wich means that that meaning dimension is salient (intensional prototypicality) to that pattern.

6.1.3 Interaction between dimensions

As we can see in Table 6.1, the interaction between the four levels of each dimension result in a 4x4 table with all but two cells filled with at least one example. Naturally, not all the combinations are equally frequent or interesting; the most salient one is certainly the collocation that indicates the prototypical context of a sense. But this does not mean that the rest of the phenomena should be ignored: we can still find interesting and useful information with other shapes of clouds, other contextual patterns, other semantic structure.

In the following sections, we will look in detail at examples of each attested combination. Each section will focus on one level of the collocational dimension, and will be further subdivided by the levels of the semantic dimension. The examples will be illustrated with scatterplots in which the colours represent HDBSCAN clusters, the shapes indicate manually annotated dictionary senses, and the transparency, the ε value from HDBSCAN. The senses are not specified in the legends, but the clusters will be named with the context word that represents it best (see Section 5.2). Textual reproductions of some tokens will also be offered; in all cases the target will be in bold face and the context words captured by the relevant model, in italics. The name of the newspaper, the date of publication and the number of the article will follow the original text, and the following paragraph will reproduce the English translation between simple inverted commas.

Table 6.1: Contingency table between the collocational and semantic perspectives, with a few examples.

Semantic interpretation	Single collocation	Lexically instantiated colligation	Semantic preference	Near-open choice
Heterogeneous clusters	*heilzaam* 'healthy/beneficial' + *werking* 'effect' (and relatives)	*herstructureren* 'to restructure' + passive aux. *word* (part of the two transitive senses); *helpen* 'to help' + *om* & *te* 'in order to'	*geestig* 'witty' + *wijze/manier* 'manner'/various adverbs; *grijs* 'grey' + colours and clothes; *herroepen* 'to recant/to void' + *uitspraak* 'statement/verdict' & juridical field	*blik* 'gaze/tin' - *werpen* 'to throw', *richten* 'to aim'
Dictionary clusters	*staal* 'sample' + *representatief* 'representative'; *schaal* 'dish of a scale' + *gewicht* 'weight'; *schaal* 'scale' + *Richter*	*herhalen* 'to repeat' + *zich* 'itself'; *hoop* 'hope/heap', in the one model that gets the senses right	*haken* 'to make trip/to crochet' + sports terms or hobby terms; *schaal* 'scale' + earthquake-topic or kitchen-topic	*huldigen* 'to honour'
(Proto)typical context	*heffen* 'to levy/to lift' and all its collocates (except for *hand/arm*); *hachelijk* 'dangerous/critical' and its collocates	*diskwalificeren* 'to disqualify' + passive aux. *word*; *helpen* 'to help' + different pronouns/prepositions (*bij*, *aan*) as only remaining context words; *herinneren* 'to remember/to remind' + *(er)aan* 'of (it)', *ik* 'I' & reflexive pron. *me*, *zich*	*grijs* 'grey' + cars; *heet* 'hot' + food; *hemels* 'heavenly' + music; *dof* 'dull' + sounds	-Not relevant-
(Proto)typical context with profiling	*stof* 'substance' and its adjectives; *horde* 'horde'	*horde* 'horde' + *journalist* & *door* 'by'	*geldig* 'valid' + tickets & dates / identity documents & *voorleggen* 'submit' /*bezitten* 'possess'; *staal* 'steel' + *ton* & *milioen* 'million' / materials	-Not relevant-

6.2 Collocation

The first level of the collocational or syntagmatic dimension is that of the collocation: clusters dominated by one context word or a group of co-occurring context words. They are most likely to be found as Cumulus clouds, but also as Stratocumulus clouds or, very rarely, Cirrus clouds.

6.2.1 Heterogeneous clouds

Albeit infrequently, collocations might transcend senses, that is, they might be frequent and even distinctive of a lemma without showing a preference for a specific sense. The most clear example is found in *heilzaam* 'healthy/beneficial', which can mean that something is literally beneficial for the health or be applied, metaphorically, to other domains as well. Its clusters tend to be dominated by one context word that is not indicative of any one sense: mostly *werking* 'effect' and *effect*, adding in some models the less frequent *invloed* 'influence'. Some examples of are shown in (5) and (6) for the 'healthy' sense and (7) and (8) for the 'beneficial' sense.

(5) Het lypoceen, een *bestanddeel* dat bijdraagt aan *de* rode *kleur*, *zou een* **heilzame** *werking hebben op de* prostaat. (*De Volkskrant*, 2003-11-08, Art. 14)

'Lypocene, a *component* that contributes to *the* red *colour*, *would have a* **healing** *power on the* prostate.'

(6) Pierik *beschrijft de* **heilzame** *effecten van alcoholgebruik op de* bloedvaten en *de* bloeddruk, op mogelijke beroerten, galstenen, lichaamsgewicht, vruchtbaarheid, zwangerschap, botontkalking, kanker, verkoudheid, suikerziekte en seniele dementie. (*NRC Handelsblad*, 1999-11-27, Art. 148)

'Pierik *describes the* **healing** *powers of alcohol consumption on* [*the*] blood vessels and [*the*] blood pressure, on potential strokes, gallstones, body weight, fertility, pregnancy, osteoporosis, cancer, the cold, diabetes and senile dementia.'

(7) Voor politici met dadendrang een gruwel, maar als men de casus van de Betuwelijn nog *voor de geest* haalt dan *zou het* advocatensysteem zijn **heilzame** *werking hebben kunnen bewijzen.* (*De Volkskrant*, 2002-03-29, Art. 79)

'For politicians with thirst for action it is an abomination, but when one recalls (lit. 'brings *to the spirit*') the case of the Betuwe line then *the* lawyer system *would* have been *able* to *prove* its **beneficial** *effect*.'

(8) *De kwestie heeft alvast* één **heilzaam** *effect*: *het profiel van* commerciële boekenprijzen staat opnieuw ter discussie. (*De Standaard*, 1999-03-27, Art. 133)

'*The matter certainly has* a **beneficial** *effect*: *the profile of* commercial book prizes is again under discussion.'

The model is shown in Figure 6.1: the clusters dominated by *werking* 'effect', *effect* and *invloed* 'influence' are shown in yellow, light blue and green, respectively, and the manually annotated senses are mapped to the shapes: the literal 'healthy' sense is coded in circles, and the general sense, in triangles. All but the *invloed* 'invloed' cluster, a Cumulus, are Stratocumulus clouds.

Within the *werking* 'effect' cluster, the literal tokens (as in (5)) are the majority and tend towards the left side of the cloud, whereas the general ones (like (7)) tend towards the right side. While there is a preference for the literal sense, especially considering that across the full sample the general sense is more frequent, it is far from homogeneous. The balance is even more striking within the *effect* cluster. Such a picture is pervasive across multiple models of *heilzaam* 'healthy/beneficial'. The vague organization within the *werking* 'effect' cluster suggests that it is not necessarily the case that the models do not capture words representative of 'physical health', but they have to compete with the most salient context words, which are not precisely discriminative of these two senses.

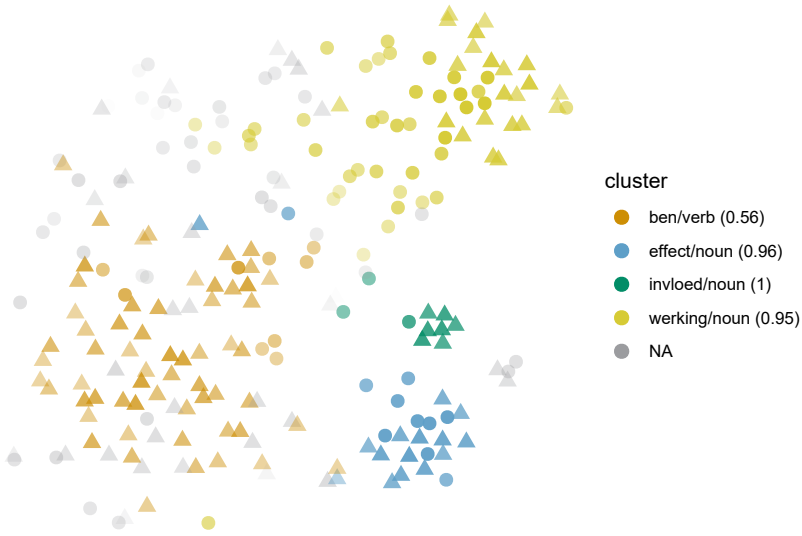

Figure 6.1: Cloud of *heilzaam*: bound10all-PPMIweight-FOCall. Circles are 'healthy, healing', triangles are 'beneficial' in general.

This is an issue if we come to the distributional semantics expecting lexical collocates, such as *werking* 'werking', *effect*, and *invloed* 'influence', to unequivocally represent different dictionary senses. On the other hand, *ben* 'to be' and *werk* 'to work, to have an effect' (of which *werking* is a nominalization), co-occur with the tokens in the orange cluster, dominated by the general sense, and less so outside this cluster; see examples (9) and (10). In other words, the most frequent nouns modified by *heilzaam* 'beneficial' tend to occur in attributive constructions (particularly *een heilzame werking hebben* 'to

have a beneficial/healing effect/power' and *de heilzame werking van* 'the beneficial/healing effect/power of') and for either sense, whereas the predicative constructions present a wider variety of nouns and a stronger tendency towards the general sense.

(9) Versterking *van de* politieke controle *op de Commissie kan* **heilzaam** *zijn maar de huidige* ongenuanceerde *discussie is gevaarlijk voor* Europees beleid en besluitvorming. (*De Morgen*, 1999-03-18, Art. 45)

'Reinforcement *of the* politicial control *at the Commission can be* **beneficial**, *but the current* unnuanced *discussion is dangerous for* European policy and decision-making.'

(10) Ten slotte nog één fundamentele bedenking: ook *de* permanente actualiteit *van de* thematiek in *de media werkt* **heilzaam** *op de weggebruikers*. (*De Morgen*, 2001-02-28, Art. 107)

'To conclude, one final fundamental thought: *the* permanent presence *of the* topic in *the media* has a beneficial effect (lit. '*works* **beneficially**') *on road users*.'

The models of *heilzaam* 'healthy/beneficial' show that that we cannot take for granted that collocations will be representative of senses. What is more, they illustrate how neither a high PMI nor their selection as cues by human annotators guarantee that a context word distinguishes predefined senses, given that these words have both a high PMI with *heilzaam* 'healthy/beneficial' and were often selected as cues by the annotators (recall Tables 4.5 and 4.6 in Chapter 4) . When it comes to PMI, it is understandable: the measure is meant to indicate how distinctive a context word is of the type as a whole, in comparison to other types. It does not take into account how distinctive it is of a group of occurrences against another group of occurrences of the same type. When it comes to cueness annotation, however, we could have expected a more reliable selection, but apparently the salience of these context words is too high for the annotators to notice that it is not distinctive of the different senses.

6.2.2 Dictionary clouds

In a few cases we can see clusters characterized by one dominant context word that perfectly match a sense, or at least its clustered tokens. These are normally fixed expressions, at least to a degree: the definition of the sense itself may specify a required expression, such as *representatieve staal* 'representative sample'.

An interesting example is shown in Figure 6.2, a model of the noun *schaal* 'scale/dish'. In the plot, the 'scale' homonym is represented by circles ('a range of values, e.g. the scale of Richter, a scale from 1 to 5'), squares ('magnitude, e.g. on a large scale') and a few triangles ('ratio, e.g. a scale of 1:20'), whereas the 'dish' homonym is represented by crosses ('shallow wide dish') and crossed squares ('dish of a scale'). Both the 'range' and the 'dish of scale' senses, exemplified in (11) and (12), have a perfect match (or almost) with an HDBSCAN

cluster, represented by a context word with perfect F-score. All the *schaal* tokens co-occurring with *Richter* are grouped in the red Cumulus cloud, and cover almost the full range of attestations of the 'range' sense, and all the tokens co-occurring with *gewicht* 'weight' are grouped in the light blue Cumulus cloud and cover all the attestations of the 'dish of a scale' sense. The blue cloud of crosses is also an homogeneous Cumulus dedicated to the 'shallow wide dish' sense, but not dominated by a collocate, and the rest are variably homogeneous Stratocumulus clouds representing parts of the 'magnitude' sense.

(11) Wenen, Beneden-Oostenrijk en Burgenland zijn dinsdagochtend opgeschrikt door een *aardschok van 4,8 op de* **schaal** *van Richter*. (*Het Nieuwsblad*, 2000-07-12, Art. 4)

'Vienna, Lower Austria and Burgenland have been scared up on Tuesday morning by an *earthquake of 4.8 on the Richter* **scale**.'

(12) Daarom is het van belang dat Nederland zich deze week achter de VS heeft geschaard, ook al legt ons land natuurlijk minder *gewicht in de* **schaal** dan *Duitsland in* het *Europese* debat over de al dan niet noodzakelijke toestemming van de Veiligheidsraad voor militaire actie tegen Irak. (*NRC Handelsblad*, 2002-09-07, Art. 160)

'Therefore it is important that the Netherlands has united behind the US this week, even though our country has of course less influence (lit. 'places less *weight on the* **dish of the scale**') than *Germany in* the *European* debate on the potentially necessary permission of the Security Council for military action against Iraq.'

In a way, the phenomenon indicates a fixed, idiomatic expression: a combination of two or more words that fully represents a sense. However, the picture is more nuanced. First, technically, the 'range' sense can potentially occur with more context words than *Richter*. In fact, one of the examples given to the annotators is *schaal van Celsius* 'Celsius scale', as well a pattern like the one found in (13), one of the orange circles at the top of Figure 6.2. However, in the corpus used for these studies, *Celsius* does not co-occur with *schaal* in a symmetric window of 4; moreover, of the 32 tokens of this sense attested in this model, 22 co-occur with *Richter*, 3 follow the pattern from (13), and the rest exhibit less fixed patterns or the infrequent *glijdende schaal* 'slippery slope' construction. The few matching (13) are more readily clustered with other tokens co-occurring with the preposition *op* 'on', such as (14). In other words, in the register of newspapers, the 'range' sense of *schaal* is almost completely exhausted in the *schaal van Richter* 'Richter scale' expression.

(13) "Misschien deelt de computer mij op grond *van statistische* analyses *op een* **schaal** *van 1 tot 12 in* categorie 3", zegt woordvoerder B. Crouwers van de registratiekamer. (*NRC Handelsblad*, 1999-01-09, Art. 10)

'"Maybe the computer on the basis *of statistical* analyses *on a* **scale** *of 1 to 12* puts me *in* category 3", says spokesperson B. Crouwers of the registration chamber.'

(14) Die stad vormde de opmaat tot de latere *collectieve regelingen op nationale* **schaal**, stellen *de* auteurs, in navolging van socioloog prof. dr. Abram de Swaan. (*De Volkskrant*, 2003-05-03, Art. 253)

'That city was the prelude to the later *collective arrangements at national* **level** (lit. '*on* a *national* **scale**'), state *the* authors, in accordance with sociologist Prof. Dr. Abram de Swaan.'

Second, the 'dish of a scale' sense need not be used in the metaphorical expression illustrated in (12), but that is indeed the case in our data. Next to *gewicht* 'weight', these tokens also mostly co-occur with *leg* 'to lie, to place' or, in lesser degree, with *werp* 'to throw'. Even in other models, this cluster tends to be built around the co-occurrence with *gewicht* 'weight', normally excluding tokens that only co-occur with *leg* 'to lie, to place', which do not belong to the same sense any more.

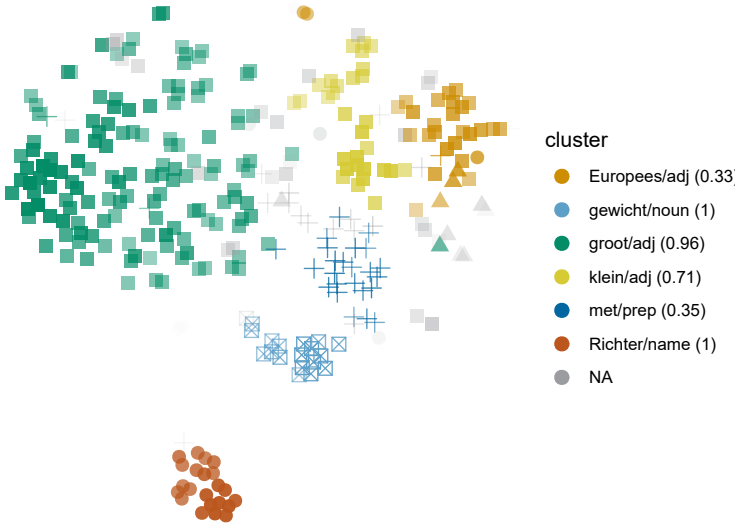

Figure 6.2: Cloud of *schaal*: nobound5all-PPMIweight-FOCall. Within the 'scale' homonym, circles are 'range'; triangles, 'ratio', and squares, 'magnitude'; for the 'dish' homonym, crosses represent 'dish' and crossed squares, 'dish of a scale'.

These examples don't disprove the possibility of clouds dominated by a collocate perfectly covering a sense, as long as we keep in mind the characteristics and limitations of the corpus we are studying and the difference between describing "how a sense is used" and "how a sense is used *in this particular corpus*".

6.2.3 (Proto)typical contexts

The most frequent phenomenon among Cumulus and Stratocumulus clouds is a cluster dominated by one context word or group of co-occurring context words that represents a (proto)typical context of a sense. It may be *the* prototypical context, if the rest of the sense is discarded as noise or spread around less clear clusters, but we might also find multiple clusters representing different typical contexts of the same sense. Neither t-SNE nor HDBSCAN can tell whether one of these contexts is more central than the other, at least not in the same sense we would expect from prototype theory. Denser areas of tokens, as perceived by HDBSCAN, are those where many tokens are very similar to each other. The more tokens are similar, and the more similar they are, the denser the area. As we will see in this example, this is not a good proxy for prototypicality.

One of the most clear examples of this phenomenon is found in *heffen* 'to levy/to lift', whose typical objects are also characteristic of its two main senses (see Figure 6.3). On the one hand, the 'to levy' sense occurs mostly with *belasting* 'tax', *tol* 'toll'[1], and *accijns* 'excise', as shown in (15) through (17). Their frequencies are large enough to form three distinct clusters, which tend to merge in the following levels of the HDBSCAN hierarchy, that is, they are closer to each other than to the clusters of the other sense. On the other hand, the 'to lift' sense occurs with *glas* 'glass', where the final expression *een glas(je) heffen op* takes the metonymical meaning 'to give a toast to' (see (18)), and with the body-parts *hand*, *arm* and *vinger* 'finger', in which they might take other metonymical meanings. The latter group does not really belong to this "collocation" category but to "semantic preference" (see Section 6.4).

(15) Op het inkomen boven *die* drie miljoen *gulden wil* De *Waal honderd procent belasting* **heffen**. (*Het Parool*, 2001-05-02, Art. 99)

'De *Waal wants* to **levy** a *one hundred percent tax* on all incomes above *that* three million *guilders*.'

(16) Mobiliteitsproblemen, rekeningrijden, op een andere manier het *gebruik* van de *weg belasten*, kilometers *tellen*, *tol* **heffen** — de *mogelijkheden om* de ingebouwde chip *te* benutten zijn vrijwel onbeperkt. (*NRC Handelsblad*, 1999-10-02, Art. 31)

'Mobility problems, road pricing, *taxing* the *use* of *roads* in a different way, *counting* kilometres, **levying** *taxes* — the *possibilities to* utilize the built-in chip are almost unlimited.'

(17) …in landen als Groot-Brittannië (waar de accijnzen op 742 euro per 1.000 liter liggen), Italië *en Duitsland* (*die beide accijnzen boven* de *400 euro* **heffen**) komt de *harmonisering* ten goede van de transportsector. (*De Morgen*, 2002-07-25, Art. 104)

'…in countries like Great Britain (where excise duties are at 742 euros per 1,000 liters), Italy *and Germany* (*both of which* **levy** *excise duties above 400 euros*) the transport sector benefits from the *harmonization*.'

[1]Typical of the Netherlandic sources, since tolls are not levied in Flanders.

(18) Nog *twaalf* andere deelnemers *konden maandagavond het glas* **heffen** *op* de *hoogste winst*. (*De Standaard*, 2004-10-20, Art. 150)

'*On Monday night* another *twelve* participants *could* **raise** *their glasses to* the *highest profit*.'

As we can see in Figure 6.3, the model is very successful at separating the two senses and the clusters are semantically homogeneous: the most relevant collocates of *heffen* 'to levy/to lift' are distinctive of one or the other of its senses. Crucially, no single cluster is even close to covering a full sense; instead, each of them represents a prototypical pattern that stands out due to its frequency, internal coherence and distinctiveness. It seems reasonable to map the clusters to prototypical patterns because of their frequency and distinctiveness, but we should be careful about how we apply the results of the modelling to this kind of semantic analysis. From the perspective of prototype theory, a feature of a category is more central if it is more frequent, i.e. it is shared by more members, while a member is more central if it exhibits more of the defining features of the categories. As such, within the 'to levy' sense, the *belasting heffen* 'to levy taxes' pattern is the most central, and tokens instantiating such a pattern will be more central. In contrast, HDBSCAN prioritizes dense areas, that is, groups of tokens that are very similar to each other. Thus, membership probabilities, which we might be tempted to use as proxy for centrality, indicate internal consistency, lack of variation. From such a perspective, given that *belasting heffen* 'to levy taxes' is more frequent and applies to a wider variety of contexts than the other two patterns of 'to levy', its area is less dense, and its tokens have lower membership probabilities within a compound of 'to levy' clusters. In other words, the models can offer us typical patterns of a lemma and of its senses and tell us how distinctive they are from each other and how much internal variation they present. Beyond this information, they don't map in a straightforward manner to our understanding of prototypicality.

It must be noted that clusters defined by collocations may not be just characterized by one single context word, but by multiple partially co-occurring context words. A clear example is *hachelijk* 'dangerous/critical', where both senses are characterized by prototypical contexts, exemplified in (19) through (24): *onderneming* 'undertaking', *zaak* 'business' and *avontuur* 'adventure' for the 'dangerous, risky' sense, *moment*, *situatie* 'situation', and *positie* 'position' for the 'critical, hazardous' sense. A model is shown in Figure 6.4, where only the yellow, orange and green clusters are Cumulus clouds, and the rest, Stratocumulus. These six frequent context words are paradigmatic alternatives of each other, all taking the slot of the modified noun, i.e. the entity characterized as dangerous or critical. However, unlike its very near type-level neighbour *situatie* 'situation', *positie* 'position' may also co-occur with *bevrijd* 'to free' (and *uit* 'from') and, additionally, with *brandweer* 'firefighter', typically in Belgian contexts. The frequency of these co-occurrences in the sample, next to the type-level dissimilarity between these three lexical items, splits the co-occurrences with *positie* 'position' in three clusters (in light blue, green and red in Figure 6.4), based on these combinations.

(19) Het is geen gewaagde stelling dat de deelname van de LPF aan de *regering*

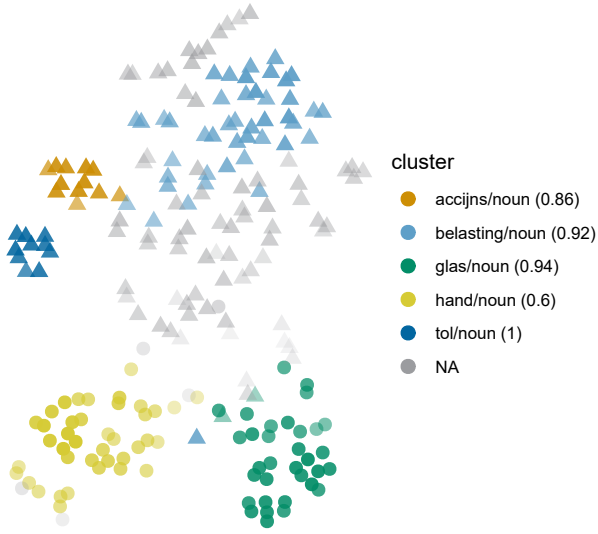

Figure 6.3: Cloud of *heffen*: bound10all-PPMIweight-FOCnav. Circles are 'to lift', triangles are 'to levy'.

een **hachelijke** *onderneming blijft.* (*De Volkskrant*, 2002-08-05, Art. 46)

'It is not a bold statement that the participation of the LPF in the *government remains a* **risky** *undertaking.*'

(20) Daar baseerden de media zich op slechts één bron, en elke journalist weet dat *dat een* **hachelijke** *zaak is.* (*De Volkskrant*, 2004-05-05, Art. 42)

'The media relied on only one source, and every journalist knows that *that is a* **dangerous** *thing to do.*'

(21) …met storm opzij is het *inhalen* van *een vrachtwagen een* **hachelijk** *avontuur*… (*Het Parool*, 2000-03-17, Art. 34)

'…under sidewind conditions *overtaking a truck* is *a* **risky** *adventure*…'

(22) *Kortrijk beleefde enkele* **hachelijke** *momenten* tegen *Brussels*, dat in zijn ondiep bad bewees zijn vierde plaats in de play-offs waard te zijn. (*Het Laatste Nieuws*, 2001-05-14, Art. 375)

'*Kortrijk experienced some* **critical** *moments* against *Brussels*, who in their shallow pool proved to be worthy of their fourth place in the play-offs.'

(23) Kort maar krachtig staat er: "De **hachelijke** *situatie* van *Palestina is* vooral een interne aangelegenheid, hoewel de bezetting en de confrontatie met Israël er de context voor schept." (*De Standaard*, 2004-10-02, Art. 162)

'Short but powerful, it reads: "The **critical** *situation* in *Palestine is* mostly an internal matter, even though the occupation and the confrontation with Israel create the context for it."'

(24) Zij toont knappe filmpjes, *opgenomen vanuit* de **hachelijke** *positie* van *een* deltavlieger… (*De Morgen*, 1999-06-07, Art. 126)

'She shows outstanding videos, *taken from* the **hazardous** *position* of *a* hang glider…'

The model does not give us information about the relative centrality of the three *positie* clusters. They result from the combination of three features, and each cluster exhibits a different degree of membership based on how many of these overlapping features it co-occurs with. At the same time, they have a distinctive regional distribution. Based on this data, we might said that a prototypical context of *hachelijke posities* 'dangerous/critical positions' in Flanders is a situation in which firefighters free someone/something from them, while this core is not present, or at least not nearly as relevant, in the Netherlandic data. We might also say that the same situation is not typical of *hachelijke situaties* 'dangerous/critical situations', and this therefore presents a (local) distributional difference between two types that otherwise, at corpus level, are near neighbours.

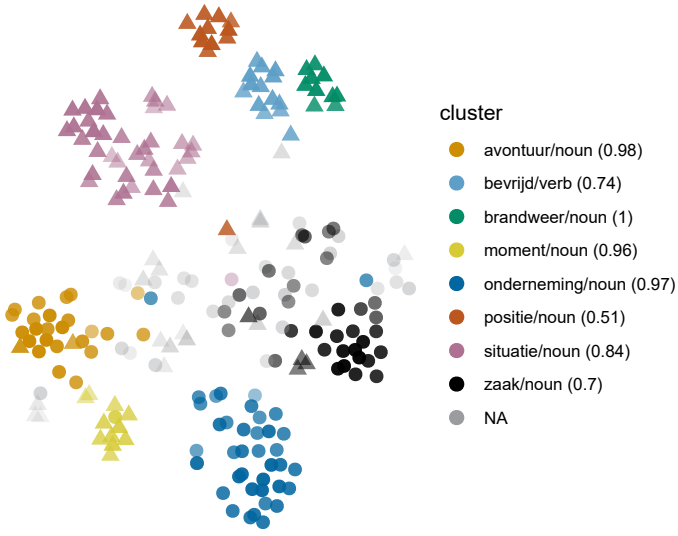

Figure 6.4: Cloud of *hachelijk*: bound5all-PPMIweight-FOCall. Circles are 'dangerous, risky'; triangles are 'critical, hazardous'.

6.2.4 Profiling

Clusters dominated by a context word may not only represent a typical context within a sense, but also one that highlights a different dimension of such sense than other clusters. This is not extremely frequent and requires an extra layer of interpretation, but it is an additional explanation to some of the clustering solutions.

One example is given by the 'substance' meaning of *stof*, represented as circles in Figure 6.5. Within this sense, we tend to find clusters dominated by *gevaarlijk* 'dangerous', *schadelijk* 'harmful' (which also attracts *kankerwekkend* 'carcinogenic') and *giftig* 'poisonous' (which often attracts *chemisch* 'chemical'). These dominant context words are nearest neighbours at type-level, and the clusters they govern belong to the same branch in the HDBSCAN hierarchy.

However, we can find additional information, among the context words that co-occur with them, which suggests that frequency is not the only responsible for their separated clusters. Concretely, the tokens in the cluster dominated by *schadelijk* 'harmful' tend to focus on the environment and composition of substances, as indicated by the co-occurrence with *uitstoot* 'emissions', *lucht* 'air', *stank* 'stench' and *bevat* 'to contain'; meanwhile, those in the cluster dominated by *giftig* 'poisonous' focus on the context of drugs or profile the liberation of substances, with context words such as *vorm* 'to form', *kom_vrij* 'to be released' and *drugs_gebruik* 'drug use'. The clusters are not distinguished by their meaning as it would be coded in a dictionary entry, but by semantic dimensions that are highlighted in some contexts and hidden in others, but always latent. This effect of the less frequent context words is one of the consequences of less restrictive models: at some levels of analysis, one word (*gevaarlijk* 'dangerous', *schadelijk* 'harmful'…) might be enough to disambiguate the target, but this extra information enriches our understanding of how the words are actually used. It is also contextualized information: not just about how *stof* 'substance' is used, but how it is used when in combination with certain frequent collocates.

6.3 Lexically instantiated colligation

Even without relying on part-of-speech tags or dependency relationships as features for our models, we can obtain grammatical information from lexical collocates. For example, the passive auxiliary *word* indicates passive constructions, as well as the somewhat less frequent preposition *door*, which indicates an explicit agent, much like *by* in English. Other constructions might also be indicated by key function words, such as *om te* 'in order to', *dat* 'that' for relative clauses, *dan* 'than' for comparatives, and prepositions. The patterns that emerge from clusters with lexically instantiated colligation may cross the boundaries of dictionary senses — resulting in heterogeneous clusters — match senses, or indicate a prototypical configuration within a sense. The following subsections explore examples of these different phenomena.

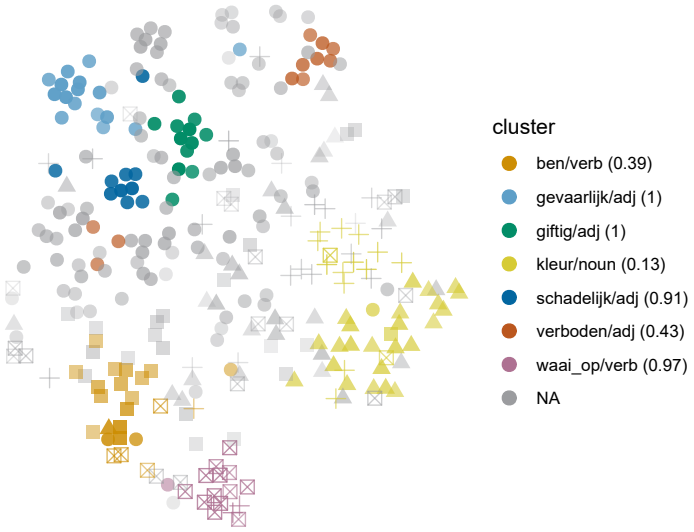

Figure 6.5: Cloud of *stof*: bound5lex-PPMIselection-FOCall. Within the first homonym, circles are 'substance'; triangles, 'fabric'; filled squares, 'topic, material'. For the second, crosses are literal 'dust' and crossed square, idiomatic expressions.

6.3.1 Heterogeneous clusters

The verb *herstructureren* 'to restructure' was annotated with three sense tags emerging from a combination of specialization, i.e. whether it's specifically applied to companies, and argument structure, distinguishing between transitive and intransitive *herstructureren*. The intransitive sense is always specific — companies restructure, undergo a process of restructure.

Models are typically not very successful at disentangling these three senses, or any one of them, for that matter. Instead, the clusters that emerge tend to highlight either the semantic or the syntactic dimension, disregarding the other one. The lexical items that most frequently dominate clusters of *herstructureren* 'to restructure' are the passive auxiliary *word*, *bedrijf* 'company', *grondig* 'thorough(ly)', and the pair of prepositions *om te* 'in order to', as illustrated in (25) through (27).

(25) OK-score deelt bedrijven op in tien klassen; klasse 1 blaakt van gezondheid, klasse 10 is op sterven na dood, ofwel, staat op de rand van faillissement en *moet grondig worden* **geherstructureerd**. (*Het Parool*, 2003-04-16, Art. 69)

'The OK-score divides companies into ten classes: class 1 is brimming with health, class 10 is as good as dead, or rather, stands on the edge of bankruptcy and *must be thoroughly* **restructured**.'

(26) *Ze* **herstructureerden** *het bedrijf en* loodsten het de internationale groep Taylor Nelson Sofres (TNS) binnen. (*De Standaard*, 2004-01-06, Art. 59)

'*They* **restructured** *the company and* steered it towards the Taylor Nelson Sofres (TNS) international group.'

(27) Uiteindelijk is dat de regering, want toen de crisis uitbrak nam de overheid een belang in de banken *om ze te* **herstructureren** *en uiteindelijk weer* te verkopen. (*NRC Handelsblad*, 2000-11-07, Art. 11)

'In the end that is the government, because when the crisis hit the authorities took an interest in the banks *in order to* **restructure** *them and eventually* sell them *again*.'

The two nouns never co-occur, and only occasionally co-occur with *word* or *om te*, which themselves co-occur a few times. Both *grondig* 'thorough(ly)' and *bedrijf* 'company' are good cues for the company-specific senses, but may occur with either transitive or intransitive constructions. In contrast, *word* is a good cue for transitive (specifically, passive) constructions, but may occur with either the company-specific or the general sense. Finally, *om te* may be attested in either of the three senses. The stark separation of the clusters in Figure 6.6 would seem to suggest opposite poles, but that is not the case at the semantic level. In fact, unlike Figures 6.3 or 6.4, dominated by Cumulus and Stratocumulus clouds, the clusters are merely slightly denser areas in a rather uniform, noisy mass of tokens — the green cloud is a Stratocumulus and the other two are Cirrus clouds — and would be much harder for the naked human eye to capture without HDBSCAN input. Instead, each cluster indicates a pole of contextual behaviour which itself may code a semantic dimension, in the case of the *bedrijf* 'company' cluster, or a syntactic one, as in the lexically instantiated colligation clusters.

6.3.2 Dictionary clouds

While a rare thing, we might be able to find a cluster dominated by a grammatical pattern that matches a dictionary sense. One clear case is the reflexive sense of *herhalen* 'to repeat', characterized by its co-occurrence with *zich* 'itself' in `BOW` models without part-of-speech filters (`all`) and in `REL` models, especially if `PPMIweight` is applied too.[2] In the model shown in Figure 6.7, it is the clearest cluster, the red Stratocumulus of squares at the bottom. Looking closely, we can see that it is made of two halves: a small one on the left, in which the tokens also co-occur with *geschiedenis* 'history', and a bigger one on the right, where they do not. This particular model is very restrictive: it normally captures only one or two context words per token, which is all that we need to capture this particular sense.

We expected this kind of output in other lemmas with purely reflexive senses as well, but it is not easy to achieve. In the case of *diskwalificeren* 'to disqualify', the very infrequent reflexive sense is typically (but not always) absorbed

[2] `PATH` models also capture *zich* 'itself', but somehow don't build clusters around it.

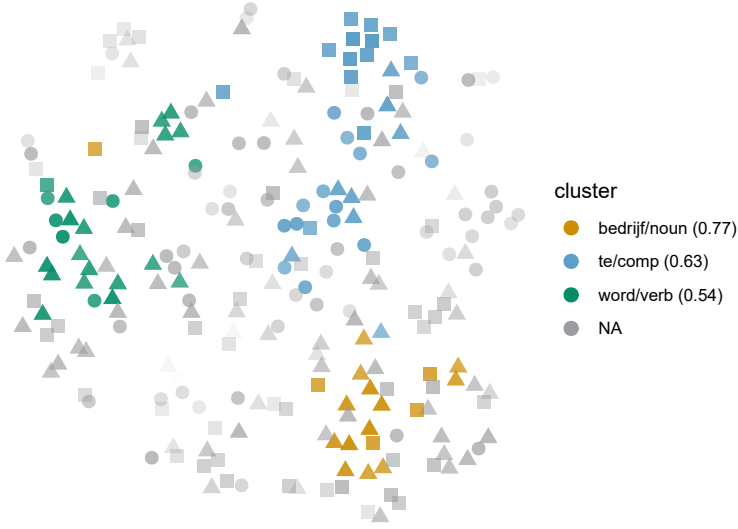

Figure 6.6: Cloud of *herstructureren*: bound3all-PPMIselection-FOCall. Circles indicate the transitive, general sense; triangles, the transitive companies-specific sense, and squares, the intransitive (companies-specific) sense.

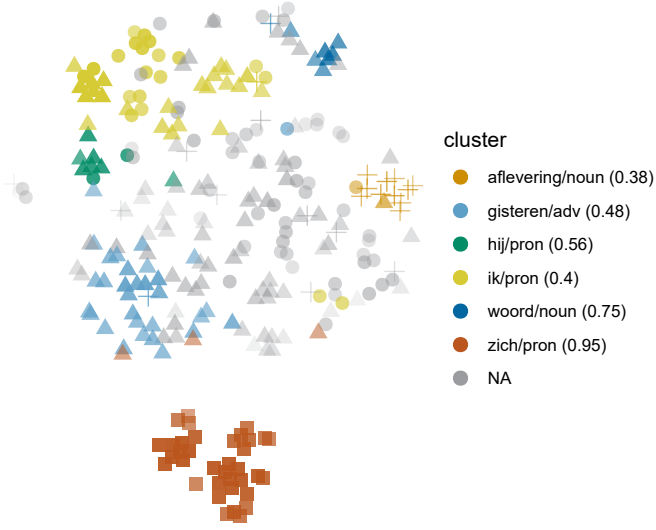

Figure 6.7: Cloud of *herhalen*: REL1-PPMIselection-FOCall. Circles are 'to do again'; triangles, 'to say again'; squares, '(reflexive) to happen again', and crosses, 'to broadcast again'.

within the transitive sense that matches it semantically, i.e. the non sports-related sense. Alternatively, a lexically instantiated colligation may prefer a certain sense without exhausting its attestations: in that case, it represents a prototypical context, as shown in the following section.

6.3.3 (Proto)typical contexts

The verb *herinneren* has two main senses defined by well defined constructions: either an intransitive construction co-occurring with the preposition *aan*, meaning 'to remind', or a reflexive construction meaning 'to remember'; a third, transitive sense is also attested but very infrequently. This lemma is sometimes rendered as three equally sized Stratocumulus clouds, as shown in Figure 6.8: the orange cluster is characterized by the preposition *aan* (see (28)), the green one by the subject and reflexive first person pronouns *ik* and *me* (see (29)), and the yellow one by the third person reflexive pronoun *zich* (see (30)). A smaller group of tokens co-occurring with *eraan*, a compound of the particle *er* and *aan* (see example (31), where it works as a placeholder to connects the preposition to a subordinate clause), may form its own Cumulus cloud, like the light blue one in Figure 6.8, or be absorbed by one of the larger ones.

(28) Vinocur **herinnert** *aan* een tekening van Plantu in L'Express. (*Het Parool*, 2002-05-18, Art. 101)

'Vinocur **reminds** [the spectator] *of* a drawing by Plantu in L'Express.'

(29) *Ik* **herinner** *me* een *concert waarop hij* hevig gesticulerend applaus in ontvangst kwam nemen. (*Het Parool*, 2003-11-14, Art. 79)

'*I* **remember** a *concert in which he* received a round of overwhelming applause.'

(30) "Het was *die dag bloedheet*", **herinnert** de *atlete uit Sint-Andries zich nog levendig*. (*Het Nieuwsblad*, 2001-08-08, Art. 192)

'"It was *scorching hot that day*", **remembers** the *athlete from Sint-Andries vividly*.'

(31) In *zijn voorwoord* **herinnert** Manara *eraan dat* deze *meisjes* in hun *tijd vaak* met toegeknepen oogjes werden aanschouwd. (*De Morgen*, 2001-11-10, Art. 40)

'In *his preface* Manara **reminds** [the reader] *that* back in their *time* these *girls* were *often* looked at with squinted eyes.'

As the shape coding in the plot indicates, the clusters are semantically homogeneous[3], because these function words are perfect cues for the senses. The rest of the co-occurring context words do not make a difference: they are not strong enough, in the face of these pronouns and prepositions, to originate

[3]With the exception of three tokens in the first-person cluster also co-occurring with *aan*, and one instantiating *ik zal herinnerd worden als* 'I will be remembered as'.

further salient structure. Nonetheless, both the *aan* and *eraan* clusters on one side, and the pronoun-based clusters on the other, belong to the same sense. Thus, what these lexically instantiated colligation clusters represent is a typical or salient pattern within each sense.

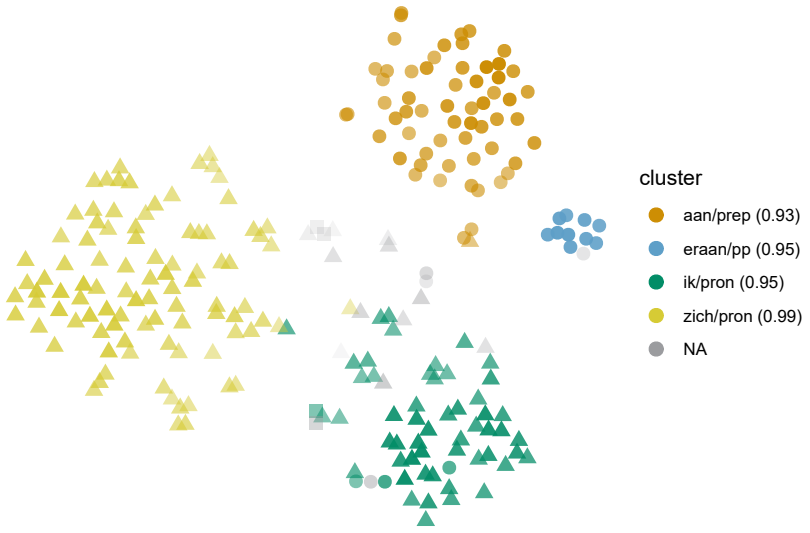

Figure 6.8: Cloud of *herinneren*: bound10all-PPMIweight-5000nav. Circles indicate 'to remind' (with *aan*); triangles, '(reflexive) to remember', and (the very few) squares, '(trans.) to remember'.

6.3.4 Profiling

Like clusters defined by collocations, clusters defined by lexically instantiated colligations can also represent a typical context that highlights a specific dimension of the sense of the target. One such case is found in the 'horde' sense of *horde*, whose most salient collocates in this corpus are *toerist* 'tourist' and *journalist*. The two collocates are quite similar to each other at type-level, but the rest of the context words in their clusters point towards a different dimension of the 'horde' sense: hordes of journalists, photographers and fans (other nouns present in the same cluster) will surround and follow celebrities, as suggested by the co-occurrence of *omring* 'to surround', *wacht_op* 'to wait' and *achtervolg* 'to chase', among others. In contrast, hordes of tourists will instead flood and move around in the city, with words such as *stroom_toe* 'to flood' and *stad* 'city'. As it stands, the situation is equivalent to the case of *stof* 'substance' described above. However, in the models that capture function words like the one shown in Figure 6.9, the profiling in these clusters is strengthened by lexically instantiated colligations. The *journalist* cluster is dominated by the preposition *door*, which signals explicit agents in passive constructions;

the passive auxiliary *word* also occurs, albeit less frequently. Meanwhile, the *toerist* 'tourist' cluster includes tokens co-occurring with *naar* 'towards'. The prepositions are coherent with the dimensions of 'horde' highlighted by each of the clusters, i.e. aggressivity and flow respectively. Interestingly, they don't co-occur with all the tokens that also co-occur with *journalist* and *toerist* 'tourist' respectively, but the nouns and prepositions complement each other instead.

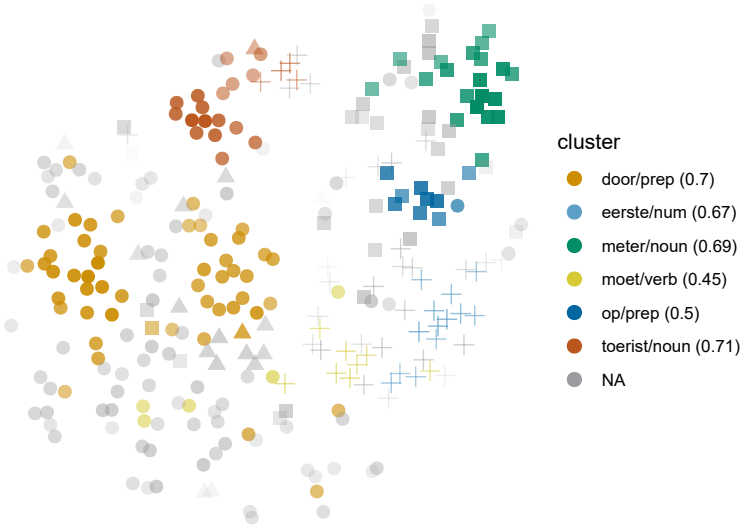

Figure 6.9: Cloud of *horde*: bound5all-PPMIselection-FOCall. Within the 'horde' homonym, circles indicate human members and triangles, nonhuman members; within the 'hurdle' homonym, squares show the literal sense and crosses, the metaphorical one.

6.4 Semantic preference

Clusters that are not clearly dominated by one context word or group of co-occurring context words, be they lexical collocations or lexically instantiated colligations, may still be the result of coherent distributional and semantic patterns. Representing first-order context words with their type-level vectors allows infrequent near neighbours to join forces and approximate the effect of one context word with their cumulative frequency. These context words may occur one to four times in the sample, that is, in about one every hundred occurrences of the target, but together with other similar context words, they form a visible pattern.

6.4.1 Heterogeneous clusters

Just like we can have clusters dominated by one context word that is not characteristic of one sense, we can have clusters dominated by multiple similar context words that are not characteristic of any sense. This is the case of names of colours and clothing terms[4] co-occurring with *grijs* 'gray', which in a model like the one shown in Figure 6.10 also includes *haar* 'hair'. As a result, *grijs* 'gray' tokens referring to concrete grey objects in general and, specifically, to grey/white hair, form the light blue Stratocumulus cloud on the top right of the figure. Note that, visually, the two senses occupy opposite halves of this cluster: the *haar* 'hair' tokens (squares) occupy their own space, but the type-level similarity of the context word to the names of colours and clothing terms makes them indistinguishable to HDBSCAN.

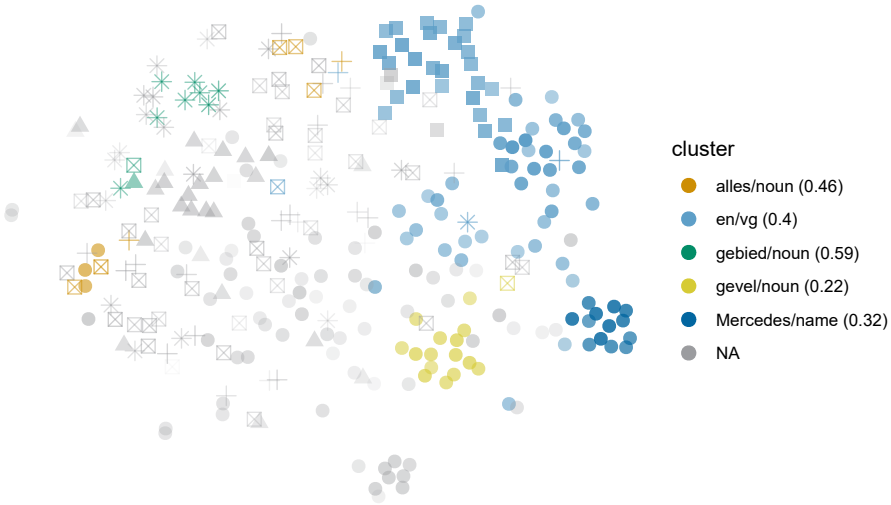

Figure 6.10: Cloud of *grijs*: bound5all-PPMIno-FOCall. Circles represent the literal sense; triangles, 'overcast'; squares and crosses, to applications to hair and white-haired people respectively; crossed squares, 'boring', and asterisks, 'half legal'.

A second example is the set of juridical terms in *herroepen*, which means 'to recant' when the object is a statement or opinion, and 'to annul, to void' when it is a law or decision. In the *QLVLNewsCorpus*, it is often used in a broad legal or juridical context. However, one of the most frequent collocates of *herroepen* within this field is *uitspraak*, which can either mean 'verdict', therefore invoking the 'to void' sense like in (32), or 'statement', to which 'to recant' applies, like in (33). Unfortunately, the broader context is not clear enough for

[4] A similar group of context words is responsible for joining the 'fabric' and 'lit. dust' senses of *stof*, even across homonyms.

the models to disambiguate the appropriate meaning of *uitspraak herroepen* in each instance. At the type-level, *uitspraak* is very close to a number of context words of the juridical field, namely *rechtbank* 'court', *vonnis* 'sentence', *veroordeling* 'conviction', etc. Together, they constitute the semantic preference of the light blue Stratocumulus cloud in Figure 6.11, which, similar to the *grijs haar* 'gray/white hair' situation above, is visually split between the tokens co-occurring with *uitspraak* and those co-occurring with the rest of the juridical terms.

(32) *Het* beroepscomité **herriep** *gisteren de uitspraak* van de licentiecommissie en besliste om KV Mechelen toch zijn licentie te geven. (*De Standaard*, 2002-05-04, Art. 95)

'*Yesterday the* court of appeal **voided** *the verdict* from the licencing committee and instead decided to grant KV Mechelen a licence.'

(33) Onder druk van Commissievoorzitter Prodi heeft Nielson verklaard dat hij verkeerd is geïnterpreteerd, maar hij heeft *zijn uitspraak niet* **herroepen**. (*NRC Handelsblad*, 2001-10-04, Art. 79)

'Under pressure from committee chairman Prodi, Nielson declared that he had been misinterpreted, but he did *not* **recant** *his statement*.'

The result is understandable and interpretable: the context words co-occurring with the tokens in the light blue cluster belong to a semantically coherent set and are distributional near neighbours. The problem is that, in the sample, the sense of *uitspraak* that occurs the most is not the juridical one like in (32) but 'statement' like in (33), therefore representing a different sense of *herroepen* than its juridical siblings. In some models, the two groups are split as different clusters, but in those like the one shown in Figure 6.11, they form a heterogeneous cluster generated by semantic preference.

Interestingly, *verklaring* 'statement' and *bekentenis* 'confession' could be considered part of the same semantic field as well, in broad terms. However, they belong to a different frame within the same field of legal action — a different stage of the process — and, correspondingly, their type-level vectors are different and they tend to represent distinct, homogeneous clusters (the green Cumulus in the figure).

6.4.2 Dictionary clusters

A few senses can be completely clustered by groups of similar context words. One of these cases was already discussed in the context of *schaal* 'scale' tokens: in models that exclude *Richter* because of its part-of-speech tag *name*, the tokens co-occurring with it can alternatively be grouped by *kracht* 'power', *aardbeving* 'earthquake' and related context words. As in the case of *Richter* as dominating collocate, the semantic field of earthquakes is not part of the definition of the 'range' sense of *schaal*, but the dominating semantic pattern within the corpus under study.

Another example is found in *haken*, where the 'to make someone trip' sense is characterized by a variety of football-related terms (*strafschop* 'penalty kick',

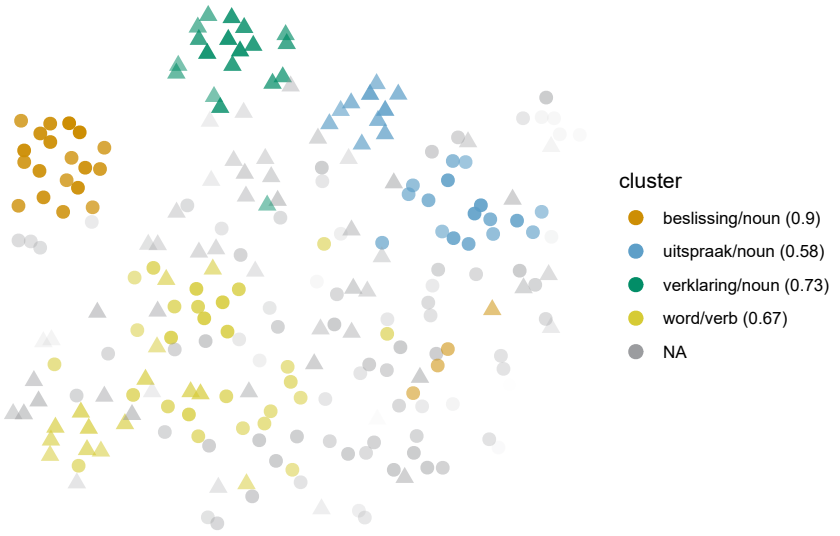

Figure 6.11: Cloud of *herroepen*: bound3all-PPMIselection-FOCall. Circles represent 'to void'; triangles, 'to recant'.

penalty, *scheidsrechter* 'referee', etc.), and the very infrequent 'crochet' sense, by *brei* 'to knit', *naai* 'to sew', *hobby* and similar words. They are represented as a Stratocumulus of dark blue squares and a Cirrus of light blue crossed squares in Figure 6.12 respectively. As indicated by the name of the dark blue cluster, the passive auxiliary *word* is also characteristic of the 'to make someone trip' cluster and very rarely occurs outside of it: here, lexically instantiated colligation is working together with the clear semantic preference of the cloud.

6.4.3 (Proto)typical contexts

There are several examples of clusters defined by semantically similar infrequent context words representing typical contexts of a sense. In Figure 6.10, for example, the dark blue Stratocumulus is represented by cars, mostly indicated by *Mercedes* and *Opel*, next to other brands. In the case of lemmas like *dof* 'dull', some models might dedicate different clusters to specific collocates, such as *klink* 'to sound', *knal* 'bang', *klap* 'clap' and *dreun* 'pounding', while others group them together in one large cluster defined by a semantic preference indicative of a sense, e.g. sounds.

A typical semantic group attested in different lemmas is culinary: found with *schaal* 'dish' — the blue Cumulus of crosses in Figure 6.10 — and with *heet* 'hot', the red Stratocumulus of mostly circles in Figure 6.13. In the case of *heet* 'hot', almost all the tokens co-occurring in this cluster refer to literally hot foods and drinks, although the full expression might be idiomatic, like in (34), and only a few of them belong to the much less frequent sense 'spicy'. In other

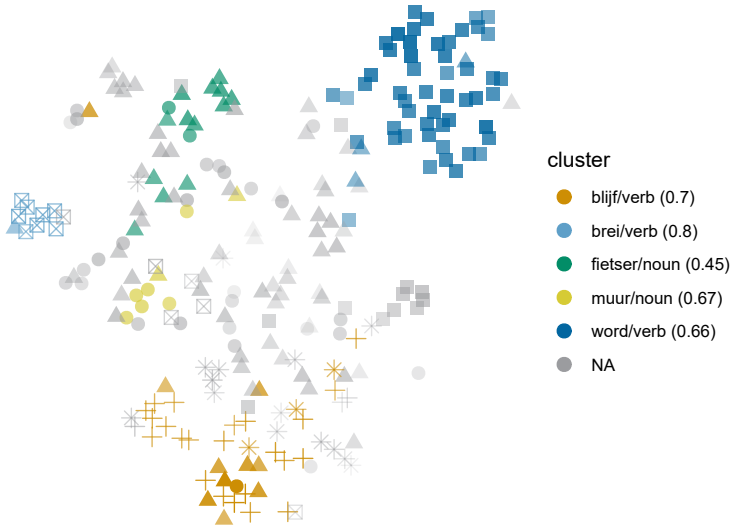

Figure 6.12: Cloud of *haken*: bound3all-PPMIselection-FOCall. Circles and triangles represent the transitive and intransitive literal 'to hook'; crosses represent the figurative (intransitive) sense; filled squares represent 'to make someone trip'; crossed squares, 'to corchet', and asterisks, 'to strive for' (with *naar*).

models, the tokens co-occurring with *soep* 'soup' and/or those co-occurring with *water* tokens might form separate clusters.

(34) Hoogstwaarschijnlijk zal Poetin Ruslands afgeknapte westerse partners discreet laten weten dat zodra hij eenmaal in het Kremlin *zit*, *de soep minder* **heet** *gegeten zal worden*. (*De Volkskrant*, 1999-12-21, Art. 22)

'Most probably Putin will discretely let Russia's former western allies know that as soon as he *is* in the Kremlin, things will look up (lit. '*the soep will be eaten less* **hot**')'

In addition, *aardappel* 'potato' is at type-level a near neighbour of the context words in this semantic group, but it still tends to form its own cluster, like the orange Cumulus in the figure. This is due both to its frequency and the distinctiveness of its larger cotext, e.g. the co-occurrence with *schuif_door* 'to pass on'. Like other expressions annotated with the 'hot to the touch' sense (circles in the figure), including *hete hangijzer* 'hot irons' in yellow and *hete adem (in de nek)* 'hot breath (on the neck)' in light blue, *hete aardappel* 'hot potato' is used metaphorically. In the strict combination of adjective and noun, the meaning of *heet* proper is still 'hot to the touch': it is the combination itself that is then metaphorized (for a discussion see Geeraerts 2003). The context words themselves are frequent and distinctive enough to generate clusters of their own with the tokens that co-occur with them, but *aardappel* 'potato' tends to stick close to the culinary cluster or even merge with it.

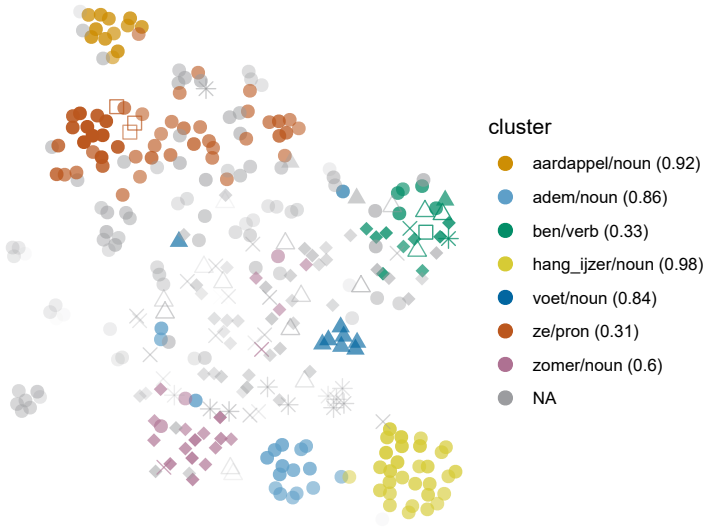

Figure 6.13: Cloud of *heet*: bound5all-PPMIno-FOCall. Among the literal senses, cricles, filled triangles and filled diamonds represent tactile, weather and body senses; empty squares and triangles represent 'spicy' and 'attractive' respectively; crosses represent 'conflictive', and asterisks, 'popular or new'.

6.4.4 Profiling

The adjective *geldig* 'valid' can relate to a legal or regulated acceptability, which is its most frequent sense in the sample, or may have a broader application, to entities like *redenering* 'reasoning'. By definition, and like for most of the lemmas studied here, each sense matches some form of semantic preference. In addition, models of this lemma reveal semantic preference patterns within the frequent, specific sense, each of which, in turns, highlights a different dimension of this sense. These patterns may be only identified as areas in the t-SNE plots or, in models like the one shown in Figure 6.14, as clouds.

The green Stratocumulus is characterized by context words such as *rijbewijs* 'driving license', *paspoort* 'passport' and other forms of identification, as well as verbs like *leg_voor* 'to present', *heb* 'to have' and *bezit* 'to possess'. In other words, it represents contexts in which someone has to demonstrate possession of a valid identification document, as shown in (35). The light blue Cirrus and the yellow Cumulus, on the other hand, co-occur with other kinds of documents (*ticket*, *abonnement* 'subscription'), *euro*, the preposition *tot* 'until', and times (*maand* 'month', *jaar* 'year', numbers, etc.). In this case, the price of the documents and the duration of their validity are more salient, as illustrated in (36).

(35) Aan de incheckbalie *kon* de *Somaliër echter* geen **geldige** *papieren voorleggen*. (*Het Laatste Nieuws*, 2001-08-24, Art. 64)

'*But* the *Somali could* not *show* any *valid papers* at the check-in desk.'

(36) Klanten van Kunst In Huis zijn bovendien zeker van variatie: wie lid is, kan elke maand een ander werk uitkiezen, het *abonnement blijft* een leven *lang* **geldig** en de *maandelijkse huurprijs* van 250 *frank is ook niet* bepaald hoog te noemen. (*De Standaard*, 1999-05-29, Art. 41)

'Moreover, customers of Kunst In Huis (lit. 'Art At Home') are guaranteed variation: members can choose a different work each month; the *subscription remains* **valid** for a life*time* and the *monthly fee* of 250 *franks is not* particularly high *either*.'

Figure 6.14: Cloud of *geldig*: bound10lex-PPMIselection-FOCall. Circles represent the specific sense and triangles, the general one.

6.5 Near-open choice

The clouds described up to now in this chapter can be easily interpreted in terms of dominating context words or semantic domains. We would expect this always to be the case: if HDBSCAN identifies a cluster, there must be structure; if there is structure, there must be an underlying pattern; if there is an underlying pattern, it can be meaningfully interpreted. Unfortunately, this is not always the case. HDBSCAN clusters can also be formed in opposition: as we saw before in the case of the Cumulonimbus clouds, i.e. the massive clusters covering at least half the sampled tokens, the grouping criterion might be a negative definition. There is a strong pattern, and everything else that does not conform to it is dumped together. In other situations, whatever structure the

HDBSCAN picks up on is very faint, compared to the Cumulus skies we may find in *heffen* and *hachelijk* (see Section 6.2.3). At present, we do not understand the relationship between HDBSCAN and token-level distributional models well enough to make sense of why these less interpretable clusters emerge and how meaningful they really are.

One of the possible interpretations of these kinds of clusters, from the linguistic point of view, is that some patterns are closer to the "open choice" side of the spectrum, while the cases discussed in Section 6.2 are closer to the "idiom" side. The open-choice and idiom principle were not really presented as poles of a continuum, but they do help as interpretative tool to make sense of the variation in cloud shapes within a lemma and across lemmas. We cannot split the data studied here between models that follow the idiom principle and those that don't, because the degree to which the distributional behaviour of each lemma can be explained by the idiom principle is different. When we generate a list of collocations for an item, we see the most relevant patterns; when we read sorted concordances, we focus on the similarities that stand out; with token-level distributional models, instead, we can see how strong or weak these patterns are.

In this section we will look at examples of clusters that cannot be interpreted in terms of dominating context words or semantic domains. Most of these result in heterogeneous clusters, especially Cumulonimbus clouds, but they can also, occasionally, bring together all the tokens of senses with certain characteristics. What I have not found is cases of near-open choice clusters that represent semantically homogeneous prototypical contexts.

6.5.1 Heterogeneous clusters

The most common situation in clusters that are not explained by a dominant context word or semantic preference, especially when they are Cumulonimbus clouds, is that they are semantically heterogeneous. These massive clouds occur in models where a small number of tokens that are very similar to each other — typically idiomatic expressions, but not necessarily — stand out as a cluster, and everything else either belongs to the same massive cluster or is noise. In many cases there is barely any noise left, while in others HDBSCAN does seem to find a difference between the many, varied tokens in the Cumulonimbus clouds and those that are left as noise.

One such example is the Cumulonimbus cloud of *blik* in Figure 6.15, shown in orange. The small Cumulus clouds to either side are represented by the co-occurrence of *werp* 'to throw' and *richt* 'to aim', which indicate prototypical instances of *blik* 'gaze' (see (37) and (38)). Very few tokens are excluded as noise — the patterns they form seem to be too different from the clustered tokens to merge with them, but too infrequent to qualify as a cluster on their own.

(37) Op zaterdag 27 april zwaait de lokale politie van de zone Kortrijk-Kuurne-Lendelede de deuren wijd *open* voor *al wie een* **blik** wil *werpen achter de schermen* van het politiewerk. (*Het Laatste Nieuws*, 2002-04-23, Art. 54)

'On Saturday 27 April the local police of the Kortrijk-Kuurne-Lendelede zone *opens* their doors wide for *all those who* want to *have a* **look** *behind the scenes* of police work.'

(38) Maar wat is goed genoeg, zo lijkt Staelens zich *af* te vragen, *haar* **blik** *strak naar beneden gericht.* (*De Volkskrant*, 2003-09-27, Art. 170)

'But what is good enough, Staelens seems to wonder, *her* **gaze** *looking straight down.*'

The orange cluster may seem homogeneous because of the predominance of the circles, but that is simply an effect of the large frequency of the 'gaze' sense, which can also occur in contexts like (39). The other sense of the 'gaze' homonym, 'perspective', as shown in (40), and of the 'tin' homonym (see (41)), are also part of this massive heterogeneous cluster. If anything brings these tokens together, other than the fact that they normally do not match the patterns in (37) and (38), is that they typically co-occur with *een* 'a, an', *de* 'the', *met* 'with', *op* 'on', and other frequent prepositions, or more than one at the same time. These frequent, partially overlapping, and not so meaningful patterns bring all those tokens together and, to a degree, set them apart.

(39) Totdat *Walsh met een droevige* **blik** *in zijn ogen* vertelt dat hij het moeilijk heeft. (*Het Parool*, 2004-03-02, Art. 121)

'Until *Walsh*, *with a sad* **look** *in his eyes*, says that he's having a hard time.'

(40) IMF en Wereldbank liggen al jaren onder vuur wegens *hun* vermeend *eenzijdige* **blik** *op de* ontwikkelingsproblemen *van Afrika.* (*Algemeen Dagblad*, 2001-02-20, Art. 129)

'The IMF and the World Bank have been under attack for years because of *their* alledgedly *unilateral* **view** *on the* delevopment issues *in Africa.*'

(41) Zijn vader had *een* fabriek waar *voedsel in* **blik** werd gemaakt. (*NRC Handelsblad*, 2003-12-05, Art. 120)

'His father had *a* factory where canned food (lit. '*food in* **tin cans**') was made.'

6.5.2 Dictionary clusters

It might seem pointless to look for meaning in clusters that do not respond to either dominating context words or semantically similar context words, but for some lemmas, it might make sense. Such is the case of the model of *huldigen* shown in Figure 6.16.

Like with other transitive verbs, the senses of this lemma are characterized by the kind of direct objects they can take. When the direct object of *huldigen* is an idea or opinion, it means 'to hold, to believe': in our sample, typical cases include *principe* 'principle', *standpunt* 'point of view' and *opvatting* 'opinion' (see examples (42) through (44)). The three of them are near neighbours at

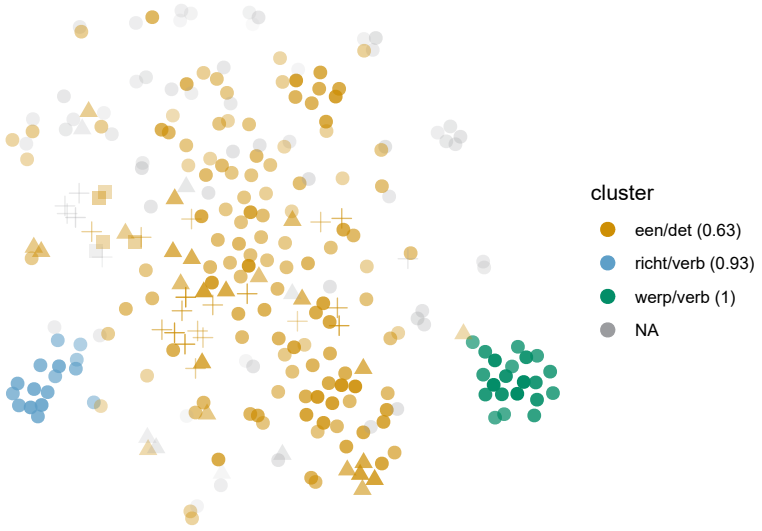

Figure 6.15: Cloud of *blik*: bound5all-PPMIweight-5000nav. For the first homonym, circles represent 'gaze' and triangles, 'view, perspective'; for the second, squares represent 'tin' and crosses, 'made of tin' or 'canned food'.

type level, but frequent enough to lead their own Cumulus or Stratocumulus clouds in most models, like in Figure 6.16. In other contexts, *huldigen* means 'to honour, to pay homage', and the role of patient is normally filled by human beings (see examples (45) and (46)). In practice, the variety of nouns that can take this place is much larger than for 'to believe', and as a result, the clusters that cover 'to honour' are less compact and defined than the clusters representing the other sense. And yet, the Cumulonimbus shown in yellow in Figure 6.16 almost perfectly represents the 'to honour' sense. How is that possible?

(42) Jacques: "Voor het *eerst* **huldigen** we het *principe* dat de vervuiler betaalt." (*De Morgen*, 1999-03-10, Art. 12)

'Jacques: "For the *first* time we **uphold** the *principle* that polluters must pay."'

(43) De regering in Washington **huldigt** het *standpunt* dat volgens Amerikaans recht de vader beslist over het domicilie van zijn minderjarige zoon. (*NRC Handelsblad*, 2000-04-03, Art. 97)

'The government in Washington **holds** the *view* that according to American law fathers decide on the primary residence of their underage sons.'

(44) …de objectieve stand van zaken in de buitenwereld zou kunnen *weerspiegelen*. Rorty **huldigde** *voortaan* de *opvatting* dat waarheid synoniem is voor wat goed is voor ons. (*De Standaard*, 2003-01-09, Art. 93)

'…would *reflect* the objective state of affairs in the outside world. *Ever since* Rorty **has held** the *opinion* that the truth is a synonym for what is good for us.'

(45) "Elk *jaar* **huldigen** wij onze *kampioenen* en sinds enkele jaren richten we een jeugdkampioenschap in", zegt voorzitter Eddy Vermoortele. (*Het Laatste Nieuws*, 2003-04-15, Art. 121)

'"Every *year* we **honour** our *champions* and for a few years we've been organizing a youth championship", says chairman Eddy Vermoortele.'

(46) Langs de versierde straten zijn we naar de kerk gereden en na de plechtigheid hebben we Karel nog **gehuldigd** in *feestzaal* Santro. Hij is nog een heel kranige man. (*Het Laatste Nieuws*, 2003-07-18, Art. 256)

'We drove through the ornate streets towards the church and after the ceremony we **honoured** Karel at the *party hall* Santro. He is still a spry man.'

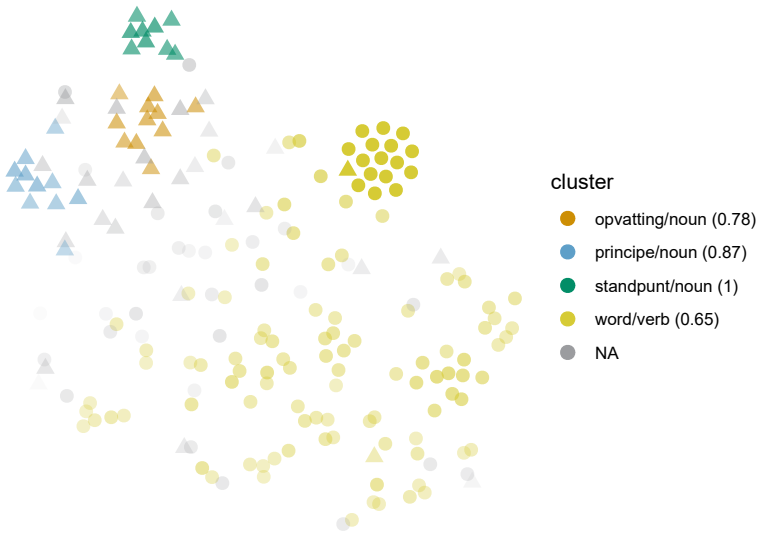

Figure 6.16: Cloud of *huldigen*: nobound3lex-PPMIselection-FOCall. Circles represent 'to believe, to hold (an opinion)'; triangles, 'to honour'.

One of the factors playing a role in the layout of this model is that the co-occurrences with *principe* 'principle', *standpunt* 'point of view' and *opvatting* 'opinion' exhaust about half the attestation of the 'to believe' sense. The rest of the tokens are too varied and typically fall into noise. The variety within the 'to honour' sense cannot compete against the stark differences between these clusters and everything else. Nonetheless, there is some form of structure within the sense that differentiates it from the equally varied remaining tokens of 'to believe', and that is a family resemblance structure.

No single semantic field is enough to cover the variety of contexts in which *huldigen* 'to honour' occurs in our sample: instead, we find different aspects and variations of the prototypical situation of ceremonies organized by sports- and city organizations in public places, in honour of successful athletes. In order to get a better picture of the syntagmatic relationships between the context words within the cluster, we can represent them in a network, show in Figure 6.17. Each node represents one of the 150 most frequent context words co-occurring with tokens from the yellow cloud in Figure 6.16, and it is connected to each of the context words with which it co-occurs in a token of that cluster. The thickness of the edges represents the frequency with which the context words co-occur within the sample; the size of the nodes summarizes that frequency, and the size of the label roughly represents the frequency of the context word among the tokens in the cluster.

The most frequent context word is the passive auxiliary *word*: it is the only context word captured in the tokens of the dense core on the upper right corner of the cloud, and co-occurs with about half the tokens of this cluster. A number of different, less frequent context words partially co-occur with it, such as *kampioen* 'champion', *stadhuis* 'city hall' and *sport_raad* 'sports council'. They subsequently generate their own productive branches in the family resemblance network. Crucially, this shows how we might have a token that co-occurs with *verdienstelijk* 'deserving' and *sport_raad* 'sports council' and one that co-occurs with *gemeente_bestuur* 'municipal administration' and *officieel* 'official', both as part of the same cluster.

Semantically and distributionally, the context words plotted in this network belong to different, loosely related fields, such as sports (*kampioen* 'champion', *winnaar* 'winner', *sport_raad* 'sports council'), town administration (*stad_bestuur*, *gemeente_bestuur* 'city administration') and temporal expressions (*jaar* 'year', *weekend*). The predominance of the passive auxiliary *word* — lexically instantiated colligation — the presence of unified semantic fields — multiple semantic preferences — and the family resemblance among tokens, resulting from an intricate network of co-occurrences, work together to model the subtle, complex semantic structure of *huldigen* 'to honour'.

6.6 Summary

Different types of clouds offer us different kinds of information. The ideal result of clusters that equal dictionary senses is only rarely found, and instead we typically find collocations that represent (proto)typical contexts within a sense. Next to this typical result, we encounter a variety of phenomena combining syntagmatic and paradigmatic aspects. Along with collocations, we find colligation and semantic preference as motors behind most of the clusters, but also a number of cases where no clear distributional pattern can be found. These phenomena correlate decently with the types of clouds discussed in Chapter 5: collocations with Cumulus clouds, lexically instantiated colligation with Stratocumulus clouds, semantic preference with all but Cumulonimbus, and near-open choice with Cumulonimbus. These are, of course, not deterministic mappings, but general tendencies. At the paradigmatic or semantic level, next

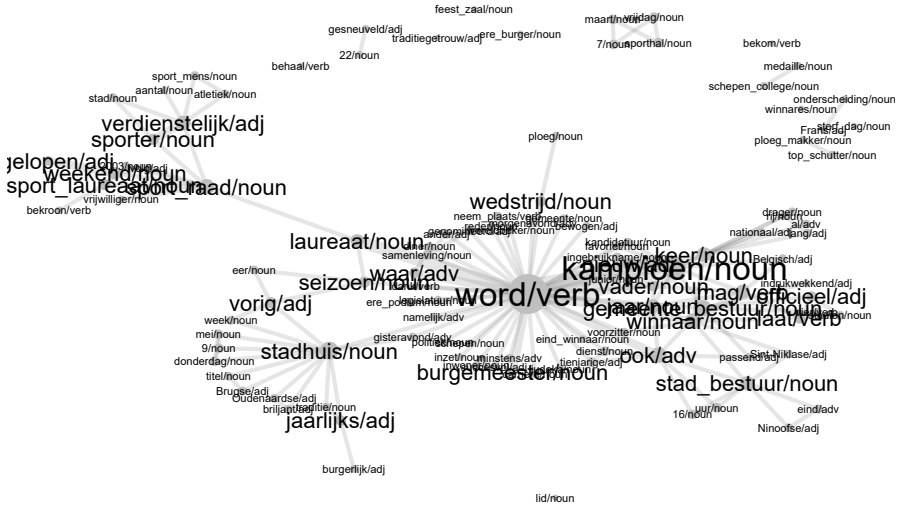

Figure 6.17: Network of context words of the *huldigen* 'to honour' cluster.

to clusters that represent typical contexts, we find heterogeneous clusters and some that match senses completely. In addition, typical contexts may include richer information regarding different semantic dimensions of a sense that are highlighted in certain contexts, i.e. that are prototypical of that contextual pattern.

In this chapter we have seen the different combinations of these syntagmatic and paradigmatic phenomena, and the shapes they can take in the models of different lemmas. Clouds do not necessarily match senses, but may offer us other types of information, depending on the distributional properties of the lemma and the dimensions that are most relevant in its semasiological structure. In the following chapter we will look at the (lack of) relationship between the information we obtain and parameter settings.

Chapter 7

No sky is the best sky

There is no magic trick to extract neat, semantically homogeneous clouds from the wild sea of corpus attestations. As we have seen in Chapter 5, the clouds can take a number of different shapes, depending on the variability of the context words that co-occur with the target, their frequency and their diversity. Chapter 6 further shows that these clusters may have various interpretations, both from a syntagmatic perspective and from a paradigmatic perspective, resulting in a diverse net of phenomena. It also explores the role of the similarity and co-occurrence between the context words. In this chapter, we will look at the relationship between these results and the parameter settings that produce them.

In consonance to the previous analyses, there is no golden law to be drawn from here. There is no set of parameter settings that reliably returns the best output: not for specific parts of speech, nor for specific semantic phenomena. This variability will be illustrated in two sections: in Section 7.1 I will compare the medoids of *hoop* 'hope/heap' and *stof* 'substance/dust…' that best model homonymy in each lemma, while Section 7.2 will look at the shape that the same parameter configuration takes in many different models.

7.1 A pile of dust

As mentioned in Chapter 4, we have modelled 7 homonymous and polysemous nouns, with the intention of studying the relationship between parameter settings and granularity of meaning. We expected certain parameters to be better at modelling differences between homonyms and others to be able to capture, at least in some cases, the more subtle differences between senses of a homonym. However, even though homonymy should be relatively easy to model[1], the results are not so straightforward. As an example, let's look at the medoids of *hoop* 'hope, heap' and *stof* 'substance, dust…' that most successfully model the manual annotation.

Figure 7.1 shows the best medoid of each of the lemmas, in terms of semantic

[1] See for example in Schütze (1998); Yarowsky (1995).

homogeneity of the clusters. By mapping the sense tags to colours, we can see that each of them has a rather well defined, homogeneous area in the t-SNE plot. It should be noted, however, that the areas are relatively uniform, and we would be hard pressed to find such a clear structure without any colour-coding. In fact, HDBSCAN only highlights the most salient areas, covering, for example, only the center of the light blue island in the left plot.

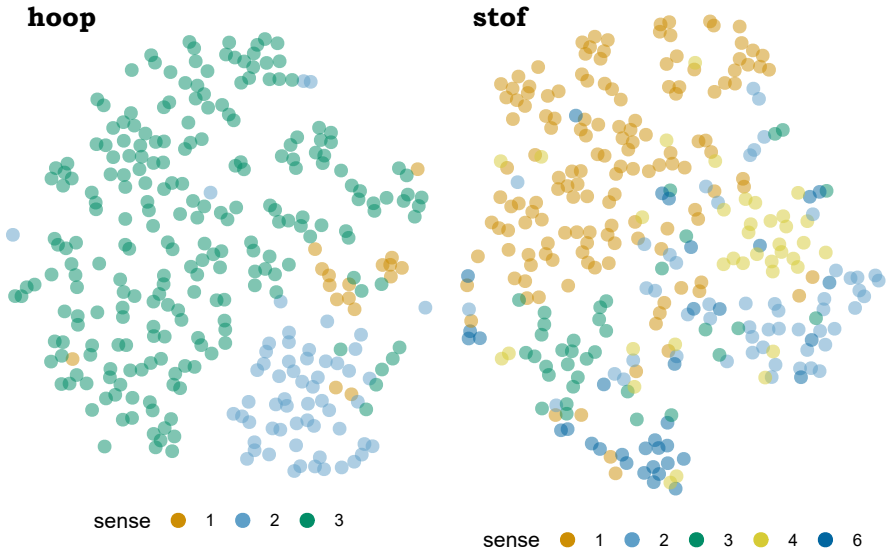

Figure 7.1: Best medoids of *hoop* (PATHweight-PPMIno-FOCall) and *stof* (bound5lex-PPMIselection-FOCall).

The senses plotted to the colours are coded with numbers to avoid cluttering. The senses of *hoop* are, for the first homonym, [1] literal 'heap, pile' and [2] general 'heap, bunch', and for the second homonym, [3] 'hope'. The first homonym of *stof* includes [1] 'substance', [2] 'fabric' and [3] 'topic, material', while the second covers [4] literal 'dust' and [6] idiomatic 'dust'. There is no sense [5], originally '(reduced to) dust', because it was not attested. Some relevant examples will be given below.

The parameters that result in these models are in fact very different, although their second-order configuration is equivalent: the union of all the context words captured by the model are also used as second-order dimensions. As a result, the dimensionality of the token-level vectors is quite low: 833 for *hoop* and 483 for *stof*.

The model that works best for *hoop* is the only medoid that manages to group the tokens of the 'heap' homonym away from the larger mass of 'hoop' tokens (in green), with even a neat moat in between. If we sacrifice the infrequent literal 'heap' sense (in orange), the split is indeed outstanding. This is achieved by a `PATHweight` model: it uses syntactic information, selects the context words connected up to three steps away from the target, and weights

the contribution of each item on that distance, regardless of the precise nature of the syntactic relationship, part-of-speech information or PMI. The syntactic distances, i.e. the number of steps to the target in the dependency path, are illustrated with the superscripts in examples (47) and (48).

In (47), the indefinite determiner *een* and the modified noun *onzin* 'nonsense' are directly linked to the target *hoop* as dependent and head respectively, so they are taken by the model and receive the highest weight. The first occurrence of the verb *is* is the head of its subject *onzin* 'nonsense', hence two steps away of the target: it is included and receives a slightly lower weight. The particle *er*, which is tagged as a modifier of *is*, and the second instance of *is*, as head of the subordinate clause, are three steps away from the target, and therefore obtain a low weight. The rest of the context is ignored by this model.

Example (48) offers a much more complex picture, particularly because the link between the target *hoop* 'hope' and the verb *spreek_uit* 'to express' (split in *sprak* and its particle *uit*), is short. As the core of the dependency tree, the main verb opens the path to many other elements in the sentence.

(47) Er[3] is[2] een[1] **hoop** onzin[1], talent is[3] niet iedereen gegeven. (*Algemeen Dagblad*, 2001-01-27, Art. 78)

'There is a **lot of** nonsense; talent is not given to everyone.'

(48) De[3] trainer[2] van[3] FC Utrecht sprak[1] verder[2] de[1] **hoop** uit[2] dat[1] hij[3] binnenkort weer eens mag[2] investeren[3] van de clubleiding. (*NRC Handelsblad*, 2004-05-24, Art. 93)

'The manager of FC Utrecht also expressed the **hope** that the club management would allow him to invest once again soon.'

A key point for this lemma is that *hoop* 'hope', represented by (48), is a mass noun, and therefore tends to occur with the definite determiner *de* (40% of the cases). In contrast, *hoop* 'heap', represented by (47), tends to occur with *een* 'a(n)' (64 out of 76 occurrences). This correlation is hard to extract with a bag-of-words model, which would either filter out function words such as the determiners, or include all determiners, related to the target or not, thus drowning this pattern in noise.

In contrast, the parameter settings that work best for *stof* are `bound5lex` and `PPMIselection`, i.e. they capture the nouns, verbs, adjectives and adverbs within 5 slots to each side of the target, as long as they are within the limits of the sentence and their PMI with the target lemma is positive. In the case of (49), for example, the model selects *discussie* 'discussion' and *lever_op* 'to bring about, to return', in italics in the transcription. Words that might follow after the period would be excluded by this model, as are those before *film* 'movie'. Within the window span of 5 words to each side, *die* 'that', *na* 'after', *veel* 'much' and *tot* 'to' are excluded because of the part-of-speech filter. Finally, the nouns *film* 'movie' and *afloop* 'end, conclusion', which survive the window size and part-of-speech filters, are excluded by the association strength filter, since their PMI value in relation to *stof* is lower than 0.

(49) Dit is een perfect voorbeeld van een film die na afloop veel **stof** tot *discussie oplevert*. (*Algemeen Dagblad*, 2003-12-11, Art. 58)

'This is a perfect example of a film that afterwards *provides* a lot of food for thought (lit. '**stuff** for *discussion*').'

Being generous, we can find a good representation of granularity of meaning for *hoop* in Figure 7.1. In the case of *stof*, however, the senses are quite well distinguished but the homonyms are not. First, most of the idiomatic 'dust' tokens group quite nicely in some sort of appendix to the main cloud. These tokens, which are by definition idiomatic uses of *stof*, tend to be very tightly grouped in most models. An example can be seen in (50). Notably, they also include a few literal tokens that also co-occur with one of the defining context words, i.e. *doe* 'to make' and *waai_op* 'to lift'.

(50) Het huwelijk tussen de hervormde Maurits en de katholieke Marylene *deed* de *nodige* **stof** *opwaaien*. (*Algemeen Dagblad*, 1999-12-08, Art. 3)

'The wedding between Maurit, a Reformed Christian, and Marylene, a Catholic, inspired a much needed debate (lit. '*stirred up* the *necessary* **dust**').'

The rest of the tokens seem to be organized by sense with subtle borders in between. The most frequent sense, 'substance', even includes a few independent islands on top, already discussed in Section 6.2.4.

Most interestingly, 'fabric' and 'dust', in light blue and yellow respectively, like to go together, even though they belong to different homonyms. In fact, HDBSCAN merges them together in one cluster, as we will see in Figure 7.3. This is not entirely surprising, given that both senses tend to co-occur with quite concrete context words, such as names for materials and colours (see for example (51) and (52)), while the 'substance' sense is more chemically-oriented and the 'topic, material' sense, illustrated in (49), co-occurs with the semantic domain of communication instead.

(51) Dankzij de nieuwe vlekwerende "stay clean"-behandelingen dringen zelfs vloeistoffen zoals olie, vruchtensap of *water* niet in de **stof**. (*De Standaard*, 2001-01-19, Art. 6)

'Thanks to the new stain-resistant "stay clean" treatments even liquids such as oil, fruit juice or *water* do not penetrate the **fabric**.'

(52) Na het **stof** de *douche*. De tocht door de Hel zit er op. (*De Morgen*, 2003-04-15, Art. 65)

'After the **dust** the *shower*. The trip through Hell [a cobblestone cycling road] is over.'

This description should suffice to understand how very different parameter configurations are necessary to model such different lemmas. The fact that both of them are homonyms is not enough: other aspects of their structure, such as the kind of contextual features that characterize each sense or homonym, play a role.

What I have not shown is that other models are not as good. What would come out from applying the parameter settings that work best for one lemma onto the other? This we see in Figure 7.2.

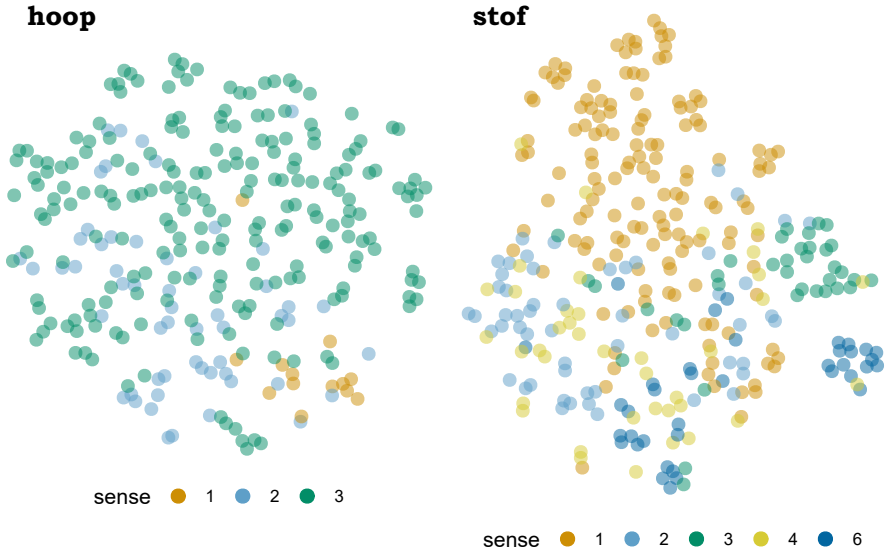

Figure 7.2: Model of *hoop* with the parameters that work best for *stof* and viceversa: bound5lex-PPMIselection-FOCall for *hoop* and PATHweight-PPMInoFOCallfor *stof*

Indeed, swapping the configurations returns unsatisfying results. In the case of *hoop*, we see a similar picture to many other models: a plot overrun by 'hope', with maybe an area with more 'literal heap' tokens, while the 'general heap' tokens, that were so nicely separated in Figure 7.1, are mixed and distributed across one hemisphere. In the case of *stof*, we keep having a large 'substance' area in orange, an isolated blue section for the idiomatic 'dust' and a shy green peninsula of 'topic, material' tokens, but the concrete senses, 'fabric' and 'dust', are disperse and mixed.

Even for a fairly straightforward task as discriminating homonyms, parameters that succeed in one lemma fail in the other. This is unrelated to the number or frequency of the senses. Instead, it is inextricably linked to the particular distributional behaviour of each lemma. While *stof* can find collocations or semantic preferences that, to various degrees, represent (parts of) senses, the lexical contexts of *hoop* are too varied to generate clear clusters. On the other hand, a syntactically informed model can identify determiners as a relevant feature of *hoop*, while the same information seems less interesting in regard to *stof*.

Table 7.1: Salient parameter settings per lemma.

	Only lex		lex or PPMIweight		No lex effect	
SOC effect	radial window	no window	radial window	no window	radial window	no window
5000-all	horde, gekleurd, hoopvol, haten, helpen	staal, blik, hemels, gemeen, grijs	stof, dof, geestig, heet			hoekig, geldig, goedkoop
5000 around	hachelijk	haken	spot, schaal	heilzaam	heffen[1]	
None	hoop, herinneren, herstellen[1], harden[1]	herhalen, diskwalificeren, herstructureren		huldigen[2]		herroepen

[1] Models with window size of 3 are separated, no radial structure.
[2] Dependency-based models are closer to those with larger window instead of those with smaller window.

7.2 Weather forecast gone crazy

Parameter settings do not have an equal effect across all models. Even at Level 1, where we compare models of a lemma with each other, we encounter a variety of patterns. Table 7.1 groups all the lemmas based on the three criteria that make the greatest difference in the organization of the Level 1 plots. The main columns refer to effects of the first-order part-of-speech filter and the PPMI weighting: in the first group of lemmas, `lex` models occupy a specific area of the Level 1 plot; in the second they are isolated next to the `PPMIweight` models (and sometimes `REL` as well), and in the third, no effect of the part-of-speech setting is found. The next level of columns distinguishes the effect of window size among the `BOW` models. A radial window configuration means that models with a window of 5 lie between those with a window of 3 and those with a window of 10. Typically, the models with smaller windows are closer to the dependency-based models, with *huldigen* being an exception. Three of these lemmas do not really exhibit a radial structure, but the models with the smallest window tend to be isolated instead. Finally, the rows indicate an effect of the second-order vectors: the first row gathers the lemmas with a separate section for the `5000all` second-order configuration; the second, lemmas where models with `5000` vectors simply have a tendency to wrap around the rest of the models (like the wings of a beetle), and the third row is used for the lemmas where second-order parameters have no special effect on the organization of their models. Models with `5000all` second-order configuration are consistently messy, and tend to make the type-level distances between all pairs of context words huge.

As we can see in the table, these patterns are not related to the part-of-speech of the target or the semantic phenomena we expect in it. This variability and the different ranges of distances between the models are the reason why selecting medoids is the most reasonable way of exploring the diversity of models.

Qualitatively, the same set of parameter settings can generate multiple different solutions, depending on the distributional properties of the lemma being modelled. We already saw this in the comparison between Figures 7.1 and 7.2: what works best for one lemma will not necessary give a decent result in another. In this section, we briefly look at the models previously plotted in Figures 5.1 and 5.2. In all cases, the parameter settings are the same of the best model of *stof*: bound5lex-PPMIselection-FOCall. The colour-coding matches the HDBSCAN clusters, and the shapes, the annotated senses.

In Figure 7.3, we see the same model for *heet* 'hot' and *stof* 'substance, dust…'. The model of *heet* 'hot' has 12 clusters, with roughly equal proportion of Cumulus, Stratocumulus and Cirrus clouds. Most of them are collocation clusters representing typical patterns within a sense, but we also find cases of semantic preference and a few heterogeneous near-open choice clusters. The *stof* 'substance, dust…' model looks roughly similar, with 7 relatively homogeneous clusters: the three Stratocumulus on the upper left are the collocation clusters discussed in Section 6.2.4 and, next to the red Cirrus defined by semantic preference, they represent typical uses of the 'substance' sense. The

rest of the clusters, as discussed above, are more heterogeneous. A further difference between the two lemmas is that, while the homogeneous clouds of *stof* 'substance' represent typical uses that profile different dimensions of the sense, the typical patterns within *heet* 'hot' constitute idiomatic expressions.

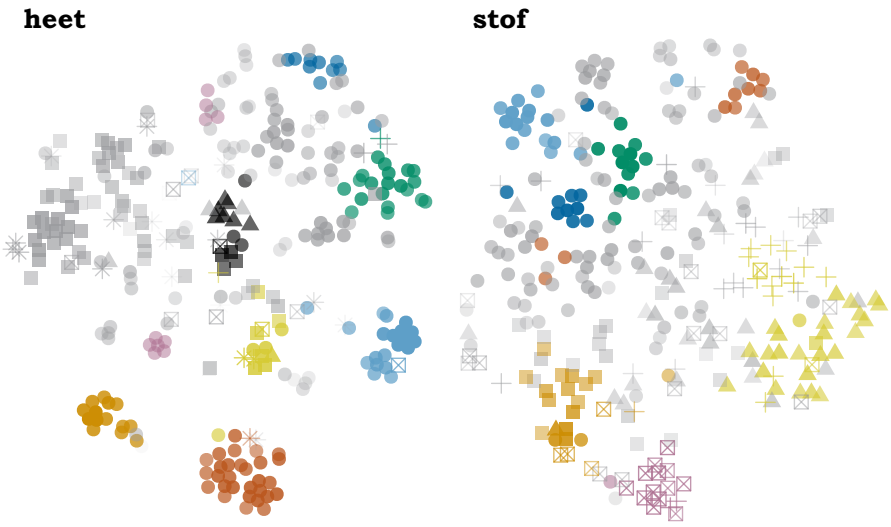

Figure 7.3: Models of *heet* and *stof* with bound5lex-PPMIselection-FOCall.

The lemmas shown in Figure 7.4, *dof* 'dull' and *huldigen* 'to believe/to honour', look rather similar to each other but very different from the ones in Figure 7.3. Even though *dof* 'dull', not unlike *heet*, tends to have multiple clusters characterized by collocations with different types of sounds, it takes a different shape in this model. The metaphorical sense represented by the collocation with *ellende* 'misery' forms a neat orange Cumulus on one side; the semantic preference for sounds gives rise to the homogeneous light blue Stratocumulus below, and the rest of the tokens, both those related to the visual sense and the rest of the metaphorical ones, gather in the heterogeneous green Stratocumulus. As we have seen before, *huldigen* also has some strong collocates, but in this model, the tokens of 'to believe', led by *principe* 'principle', *opvatting* 'opinion' and *standpunt* 'point of view', take part of an extremely homogeneous orange Stratocumulus, while most of the 'to pay homage' sense covers the light blue Cumulonimbus, like in the case described in Section 6.5.2.

The lemmas in Figure 7.5, *haten* 'to hate' and *hoop* 'hope/heap', show yet another configuration generated by the same parameter settings. Except for the green Stratocumulus in *haten*, roughly dominated by *mens* 'human, people', the rest of the clouds are Cirrus clouds: small, heterogeneous, characterized by many different words.

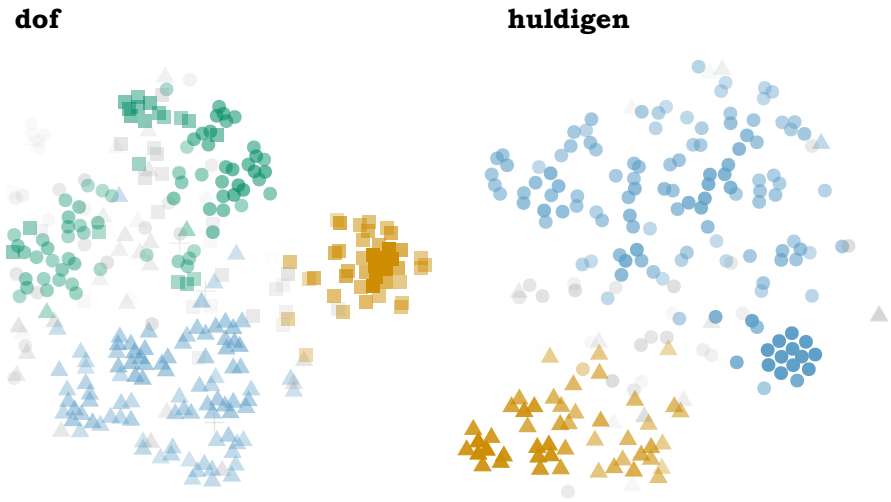

Figure 7.4: Models of *dof* and *huldigen* with bound5lex-PPMIselection-FOCall.

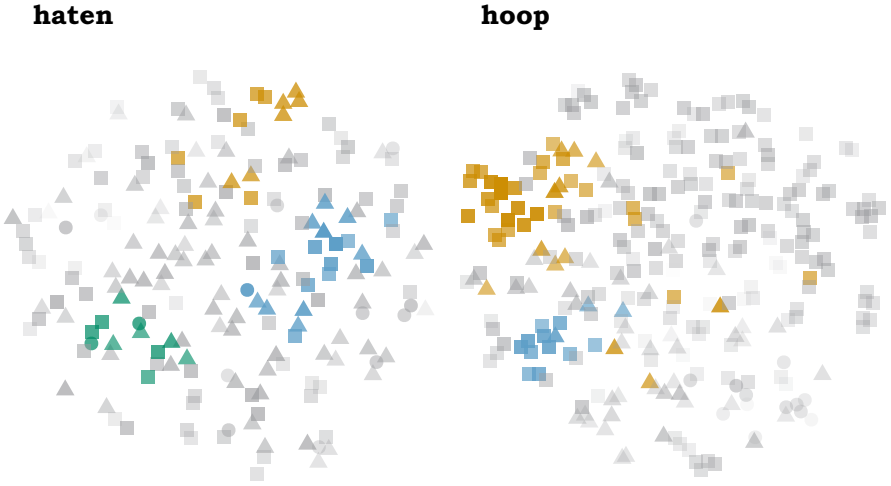

Figure 7.5: Models of *haten* and *hoop* with bound5lex-PPMIselection-FOCall.

7.3 Summary

The output of a model is not directly predictable from its parameter settings. Clouds can take many shapes, lemmas exhibit different distributional patterns, and these patterns can have different semantic interpretations. The parameter settings that model one phenomenon best, in a certain model, will not necessarily model the same phenomenon in another lemma, or anything else of interest for that matter. The same parameter settings can result in drastically different shapes across lemmas, or even if the shapes are similar and they are the result of comparable distributional behaviours, they might have different semantic interpretations.

With these cheerful thoughts, the analytical part of this dissertation comes to an end. In the next chapter I will conclude with a brief summary of the findings in the form of guidelines — tips and tricks for the interested cloudspotter — thoughts for further research.

Part III

The cloudspotter's cheatsheet

Chapter 8

Conclusions and guidelines

The focus of this dissertation is methodological: rather than describing a specific phenomenon in language, e.g. metaphorical extensions of temperature terms in Italian, it develops and tests a workflow that could be used in concrete case studies. It combines computational techniques with a Cognitive Semantics framework with the aim of implementing NLP tools to lexicological and lexicographical research. From this position, the main research questions revolve around the possible mappings between parameter settings, i.e. sets of decisions that generate different models, and semantic phenomena of lexicographic interest:

- Which parameter settings model senses the best?
- How can we tailor the parameter settings to capture homonymy, metaphor, specialization, argument structure…?

In addition, since manually annotated senses are not taken as a unique truth and, beyond accuracy, we are interested in what makes models (fail to) approximate human-based categories, the study incorporates *ad hoc* visual analytics for a fluid, quantitatively-rich qualitative analysis.

After an initial presentation of the foundations of the study in the Introduction, Part I, *The Cloudspotter's toolkit*, laid out the methodological background. Chapter 2 described the computational techniques and the methodological choices, Chapter 3 showcased the visualization tools and Chapter 4 introduced the selected lemmas and the annotation procedure.

Part II, *The Cloudspotter's handbook*, discussed the results of the analyses. Even though the answer to the original research questions is negative, it is indeed possible to learn something from the models, and these three chapters elaborate on these possibilities. Chapter 5 offered a typology of the nephological shapes, for not all the clouds in the sky are white and fluffy. These shapes result from identifiable properties of the contexts and can be interpreted in different ways. Chapter 6 followed with a systematization of these possible interpretations from a linguistic perspective. A net of phenomena is woven from a combination of paradigmatic relations — from heterogeneous clusters to clouds that reveal semantic profiling of patterns — and syntagmatic relations

— from collocations through semantic preference to open-choice tendencies. They are not the same phenomena we set out to investigate initially; although we may find metaphor, metonymy, specialization and argument structure, it greatly depends on each lemma and on how it matches these semasiological categories to its distributional behaviour. It is not enough for a lemma to *have* metaphorical extensions, they also have to correlate with salient contextual patterns. Nevertheless, we do find linguistic properties — and particularly the kind of properties that corpus-methods can capture while other empirical approaches might not. Finally, Chapter 7 illustrated the negative answer to the main question: there is no set of parameter settings that works best across the board. Each lemma has a different semasiological structure in terms of distributional behaviour, thus applying the same tool will return different results. If a parameter configuration is a cookie cutter, the various lemmas are kinds of mixtures: lemon-flavoured cookie dough, dough with chips, dough flattened by an embossed rolling pin… or even sourdough or cake batter.

In the remainder of this chapter I will summarize some points that emerge from the dissertation as a whole. First, Section 8.1 offers a possible explanation for the discrepancy between the expectations that we may come to distributional models with and the actual results. However, this shall not stop us: Section 8.2 lists a few technical guidelines for model building, based on the set of models explored here, and Section 8.3 is dedicated to general suggestions for further research based on what was not done for this project. Finally, Section 8.4 summarizes the contributions of this dissertation to distributional approaches to semantics.

8.1 Types, tokens and clouds

Distributional models rely on the Distributional Hypothesis: words that occur in similar contexts tend to be semantically similar. That seems to work for types, and projecting the intuition onto the token-level sounds straightforward: attestations occurring in similar contexts will be semantically similar, and those occurring in different contexts will be semantically different. Semantic distinctions between attestations of a word, i.e. their semasiological variation, are normally grouped as senses. So it stands to reason that we can use token-level distributional models to find senses (Schütze 1998, Yarowsky 1995). However, this line of reasoning has two issues.

On the one hand, there is the issue of patterns. At the type-level, vector representations aggregate over all the occurrences, building profiles that take into account patterns of attraction and avoidance across hundreds, thousands or even millions of events. Similar words share the same tendencies; different words prefer different things. The intuition behind distributional models is often illustrated with examples like the following (Pantel & Lin 2002: 613):

> A bottle of *tezgüno* is on the table.
>
> Everyone likes *tezgüno*.
>
> *Tezgüno* makes you drunk.

We make *tezgüno* out of corn.

The authors make the point that the words in the context of *tezgüno* suggest that it may be a kind of alcoholic beverage, because other alcoholic beverages tend to occur in similar contexts (Pantel & Lin 2002: 613). And indeed, at *type-level*, such patterns are likely to generate a distributional profile for *tezgüno* that is similar to that of *beer*, for example. And even though actual contexts are rarely as self-explanatory as these examples, type-level distributional models — to some degree at least — *work*.

Type-level models will be most similar between words with similar overall *patterns*: tendencies towards or against certain contexts. Each individual context is not enough. The examples above highlight different properties of *tezgüno*, namely that it is a liquid stored in bottles, that people have (positive) opinions about it, that it is alcoholic and that it is made out of corn. The range of items that could occur "in the same context" of *tezgüno* will depend on which of the contexts we take into account. Take, for example, the following replacements:

A bottle of *water* is on the table.

Everyone likes *you*.

Whiskey makes you drunk.

We make *cornflakes* out of corn.

Each context is not enough: at most, they set up situations in which some meaning or meaning dimension fits, while the other dimensions, whatever they are, are backgrounded and irrelevant. Type-level models work because they look at all the contexts together. At the same time, we cannot really know if the *tezgüno* that makes you drunk and the one made of corn are the same *tezgüno*; type-level models build on the assumption that they do, and for that reason they conflate semasiological structure.

In the same way, token-level models look for patterns, i.e. tendencies towards or against certain contexts or context words, but with a much more restricted pool of variables. First, the context of a token contains fewer variables than the aggregated context of a type to draw a pattern from, which results in more polarization and less nuance. Frequently co-occurring words will dominate and define what counts as a pattern, while weaker words will lack the necessary distinctiveness to impose their patterns. And because authentic concordances are not neat, propositional, explanatory descriptions of the targets, these patterns do not necessarily match *senses*.

That is, in fact, the second issue. The possibility of determining what counts as different senses is debatable (Geeraerts 1993, Glynn 2014), so why should we look for senses in the first place? Indeed, Geeraerts suggests a procedural rather than reified conception of meaning: "words are searchlights that highlight, upon each application, a particular subfield of their domain of application", and adds that "the distinction between what can and what cannot be lit up at the same time is not stable" (Geeraerts 2006: 137). In terms of clouds, context words compete for the opportunity to signal the subfield highlighted by the

target at the moment. The result is imprecise for several reasons. First, the context words are represented as type-level vectors that generalize over their most salient patterns, which are not necessarily the relevant dimension in this context, as in the case of *uitspraak herroepen* 'to recant a statement/to void a verdict' discussed in Section 6.4.1. Second, the dimension the context words highlight are not necessarily the ones we are interested in; there is structure in models of *heilzaam* 'healthy/beneficial' discussed in Section 6.2.1, but it does not correspond to the distinction between literally healthy or healing and metaphorically healthy, i.e. beneficial. Third, and in relation to the issue of patterns, the context words might be too infrequent and not distinctive enough for their voice to reach us.

On the bright side, there is so much variation across these patterns that their shapes alone are already interesting information. All words can be described with lists of collocations, but token-level models reveal how strong (or weak), how distinctive, how widespread the collocations are within the scope of the target. And beyond the clouds themselves, visualizing the models can let us see spatial organization that might be missed by clustering solutions, such as the fact that the occurrences of *uitspraak herroepen* 'to recant a statement/to void a verdict' come together while staying close to other instances of *herroepen* 'to void' in a juridical context, or the fact that health-specific and general attestations of *heilzame werking* 'beneficial effect' occupy opposite poles of the same cluster. Distributional models might not replicate our intuitions about the semantic distinctions within a lemma, but will offer us a different, complementary perspective that only they, by scanning and organizing hundreds of empirical observations, may capture.

8.2 Practical tips

Even if there is no infallible parameter settings configuration and it is hard to predict their output, some guidelines are possible. In this section I would like to offer some suggestions for a future case study that would use distributional semantics and, of course, the visualization tools presented here, to investigate the semasiological structure of a given lemma. The initial research questions would go along the lines of "How strong are the collocational patterns of this lemma?", for example. Given the variety of results from the 32 lemmas analysed for this dissertation, all these guidelines can offer is a starting point to explore the distributional behaviour of a lemma; further steps to refine the questions and fine-tune the models would depend on the results from such initial exploration. In broad terms, the outline of such a case study would be as follows:

1. Choose your lemma(s)[1]. In the Nephological Semantics project we look

[1]They could be separate words, as different case studies, or related words that might overlap in the application. In that case, it would be possible to combine a semasiological perspective, i.e. looking at the distribution of each lemma, with an onomasiological perspective, i.e. exploring the overlap and differences between the lemmas.

for ways of scaling up this procedure, but these are suggestions for small-scale studies, where a detailed examination of the clouds is viable.

2. Set up a range of parameter settings that are not too restrictive:

    - keep window sizes above 3;
    - forget about sentence boundaries;
    - avoid long, unfiltered type-level vectors;
    - don't bother with `REL` templates;

3. Generate hundreds of models on a manageable sample of tokens based on those parameters;

4. Explore the plot of models in Level 1 of NephoVis (Section 3.2) to get an idea of how the parameter settings interact;

5. Compute up to 9 medoids with PAM and explore them in Level 2 of NephoVis;

    - I chose 8 because it was the minimum that kept enough variation across lemmas, but on a lemma-by-lemma basis it could very well be reduced. More than 9 medoids are difficult to visualize simultaneously.

6. Cluster the models with HDBSCAN and explore them with the ShinyApp, finding types of clouds, collocational patterns, etc. The classifications in Chapters 5 and 6 will be useful, for example:

    - Cumulus clouds (very tight and salient) tend to be dominated by strong collocates and represent typical usages of a sense.
    - Cumulonimbus clouds (the huge ones) are normally as good as noise tokens.
    - When Cirrus clouds (the small, wispy ones) are the most salient clusters, they are capturing the little structure there is. The model is probably characterized by weak collocational patterns.

7. Interpret the clusters.

    - What are the models saying? Are there collocates, lexically instantiated colligates, semantic preference, or neither? Are the clusters heterogeneous or homogeneous? Could they be considered different senses?
    - Which medoids exhibit a more interpretable structure? What parameters do they represent?[2]
    - How much more data is left to annotate?

8. If necessary, readjust the parameters and/or incorporate manual annotation and start again.

[2]For example, via a conditional tree with the clustering solution as response and the parameter settings as predictors.

Among the interpretative questions, one of the most crucial ones is: "Could they be considered different senses?". I already mentioned in the introduction that the prototypicality of categories leads us to be sceptical about the existence of discrete senses. Accordingly, the clouds offer an alternative view on the semasiological structure of a lemma: a classification that neither matches dictionary senses nor replaces them, but could inform semantic research nonetheless. In the rest of this section I will elaborate on some of the recommendations made above.

First, I would discourage very restrictive models. We might be tempted to remove as much noise as possible and only leave context words that are very informative, which sounds reasonable in theory. But even assuming you can figure out which words are going to be informative — e.g. via annotation of cues — the result might not be what you expect. Restrictive models tend to generate clouds with Hail: dense areas with identical tokens, which override more subtle relationships. The less "relevant" context words might be harmful, but they might also make no impact whatsoever, or even add information we did not expect, like the semantic profiling of specific patterns. That said, some lemmas may require very strict settings because the context words that would then be captured are already varied enough.

Concretely, window sizes smaller than 5 tend to be too restrictive, while the window size of 10 is already bordering into too noisy. Within the dependency-based models, `RELgroup1` models are often too restrictive and rarely informative enough. A wider variety of `REL` templates is more useful, but in any case, designing the templates to fit increasingly complex patterns — especially when chains of verbs come into play — is time consuming and never good enough. `REL` models could be discarded altogether, unless the researcher has a good idea of which templates are useful for the specific lemma under study. For example, *haten* 'to hate' tends to occur in active constructions without chains of modals (e.g. *ik haat het* 'I hate it'), while *herroepen* 'to recant, to void' often co-occurs with the passive auxiliary, modals or even both (e.g. *het nachtverbod moest worden herroepen* 'the night ban had to be voided'). As a result, a simple `REL` template capturing the direct object of the verb could be enough for *haten* 'to hate'[3] but would miss many of the *herroepen* 'to recant, to void' tokens.

In a similar vein, `PPMI` can be too restrictive for some lemmas and should be used with care, especially `PPMIweight`, which might enhance the influence of already powerful context words and, for example, cause Cumulonimbus clouds. Since the filtering power of `PPMIselection` depends on the range of association strength values between the target and its context words, it is not straightforward to find a threshold that is just as restrictive as we want it to and not more for every lemma. Instead, it could be fruitful to test out different thresholds — and even combine other measures — on a lemma-by-lemma basis.

One parameter setting that should be certainly avoided is `5000all`, which often makes a great impact in the difference between models but never for the

[3]We might want to do this because in the current models the strongest context word is *ik* 'I', which does not contribute to the disambiguation. However, a brief test modelling the direct objects revealed that they were grouped thematically instead of by animacy, and thus could not model the distinction between 'to hate' and 'to dislike' either. Maybe other second-order settings could return a more adequate model.

better. Either applying a part-of-speech filter or reducing the dimensionality, e.g. by using the first-order context words as second order dimensions (`FOC`), already gives better results. This is most likely due to sparsity and/or low informativeness of the dimensions selected by `5000all`, so applying SVD afterwards might also help.

Finally, ignoring sentence boundaries does not seem to make a difference. In most cases, Level 1 plots place models that are only different on this parameter right next to each other; the few times that it makes a difference, two or three other parameters are already more important.

These tips should help in the selection of parameter settings for future models, but it is still a good idea to generate multiple models and look at their medoids. Chapter 7 showed that there is no unique recipe to tailor a model to disambiguate in a certain way. Models find patterns based on the distributional behaviour of the lemma — how frequent its context words are, how similar they are to each other, how often they co-occur, etc. The degree to which these patterns match senses in general or any sort of semasiological structure — homonymy relations, metaphor, idioms, argument structure… — is an empirical question, and that is what this procedure addresses. Fine-tuning can only be implemented after the first set of medoids have traced an outline of the lemma's structure.

What is more, the medoids can also provide an estimation of how much manual annotation is actually needed. Given a model like *heffen* 'to levy/to lift' or *herinneren* 'to remember/to remind', the patterns are so clear and homogeneous that checking the main context words of the different clusters and a few of their concordance lines is enough; at most, you would need to examine some noise tokens more closely. At the same time, in a case like *heilzaam* 'healthy/beneficial' you would immediately see that the collocation-based clouds are semantically heterogeneous, while a case like *haten* 'to hate' might make you want to rethink your life choices. In any case, you don't need to annotate all the tokens at the beginning unless there is an *a priori* classification you are intent in finding. Even then, it's best to keep it under 6 categories, or it becomes really hard to distinguish their colour-coding visually.

These suggestions should avoid a lot of trial and error in case-studies along these lines. Interpreting clouds when we have not seen any before and, especially, if we expect them all to be clearly-defined islands, is quite challenging already. Besides, as Geeraerts (2010a: 73) argues, "empirical research involves an empirical cycle in which several rounds of data gathering, testing of hypotheses, and interpretation of the results follow each other", and cloudspotting is no exception.

8.3 To the sky and beyond

The choices described in the Introduction and Chapter 2 implied leaving out the alternatives, which could very well be explored in future research projects.

At the level of parameter settings, other selections of part-of-speech filters, for example expanding `lex` with proper names and prepositions, could offer a middle point between the two options that were examined, since `lex` was

sometimes too restrictive, while `all` could be too noisy. When it comes to dependency-based models, the natural extension is to incorporate the dependency path into the feature, e.g. with "is object of *to eat*" as a feature. This is technically more challenging and likely to result in sparser vectors, but would make the connection between the target and the second-order dimensions more clear. In the current implementation, the relationship between the target token *study*$_1$ and its second-order dimension *language/n* in Table 2.2 is given by the association strength between said second-order dimension and the first-order context word *lexicography/n*: *lexicography/n* occurs in the immediate context of *study*$_1$ and has a PPMI of 4.37 with *language/n*, so the coordinate of *study*$_1$ in the *language/n* dimension is 4.37. If dependency relations are built into the feature, e.g. "its object is *lexicography/n*", the dimensions highlighted by that feature would be other verbs that take *lexicography/n* as object.

In relation to this issue, the precise effect of the second-order parameters has not been thoroughly explored, but techniques should be devised to better understand the effect of the second-order dimensions. Moreover, instead of comparing `FOC` second-order vectors with longer ones based on frequency, they could be compared with `FOC` vectors based on different samples: `FOC` models transfer the context words that survived the first-order filters as second-order dimensions, so the same set of parameter-settings on different samples — particularly on samples of different sizes — may result in different selections of context words. Additionally, they could be compared to implicit type-level vectors (Lenci 2018), i.e. where the dimensionality was reduced by SVD or non-negative matrix factorization, or even prediction-based vectors. The original reason not to implement this was to keep the transparency of the vectors to a maximum (Heylen et al. 2015), but the transition to second-order vectors already obscures the meaning of the dimensions to a great extent.

Following this reasoning, the motivation to exclude prediction-based models disappears. On the one hand, type-level word embeddings could be incorporated as representations of the first-order context words. On the other, given the possibilities offered by the family of BERT models, BERTje (de Vries et al. 2019) could be applied to the tokens themselves. For a proper comparison between the methods, new models would have to be created with word forms as units, re-tokenizing the corpus with BERTje's tokenizer. The first goal would then be to check how well the classifications presented in Chapters 5 and 6 can be mapped to models based on word forms and to what degree they also apply to BERTje models. Nonetheless, concerns about the tokenization should be addressed: the output might be useful for certain NLP tasks, but if words cannot be captured because the tokenization breaks them (as is the case of *heilzaam*, which is split between *heil* and *##zaam*), the utility of BERTje for lexicographical purposes decreases. A solution might be the implementation of larger units as targets and features in modelling procedure, such as bigrams. That in itself is another interesting avenue for further research, since words do not work in isolation, but technically more challenging.

Not only the model-building process, but also the model-analysis process could use a deeper exploration. First, the possibility of implementing UMAP should be explored. Based on initial comparisons, the clarity of the clusters

does not seem to be very different from the t-SNE output, but the shapes are different and their relative distances are supposed to be interpretable. In addition, HDBSCAN clustering with $minPts = 8$ replicates the visually identified patterns quite well, but it is not always clear when tokens are excluded as noise or how distinctive the clusters have to be to split. That said, switching to UMAP, other perplexity values for t-SNE and/or other $minPts$ values for HDBSCAN *may* void the warranty on the classifications and descriptions offered in this dissertation.

8.4 Summing up

Distributional semantics addresses an issue for descriptive linguists who would like to use corpus methods for semantic analysis. Such a linguist would be eager to exploit the increasingly large available corpora but tired of manually annotating hundreds of concordances with sense tags that might not even be that appropriate[4]. Distributional models, on the other hand, present themselves as a scalable, automatic approach that can process large amounts of textual data and extract patterns with semantic correlates. They constitute an irresistible asset for empirical approaches aiming to maximize the automation of the most laborious, quantitative tasks and give the researcher more energy and time for the creative and hermeneutic aspects of research. This dissertation was written for such a linguist, and it has good news and bad news.

The bad news is that, although distributional models can indeed reveal patterns and offer information that we might not obtain by other means, these are not necessarily *the* patterns and information we would have expected. The results from this study suggest that, if we are to use distributional semantics for descriptive analyses, we should not do so blindly. Unlike what high accuracy scores on benchmarks would suggest, there is no parameter setting that works optimally across the board, because what is relevant in the description of one lexical item might not be for another. For the same reason, different configurations of parameter settings will have different effects on each lemma, highlighting specific aspects that may be more or less interesting from a linguistic perspective. They may be senses, or they may be something else.

The good news is that a user-friendly, comprehensive visualization tool is available for the exploration of such models. Interfaces like the ones described here turn the apparent chaos of distributional models into concrete visual representations for us to examine and interrogate. Rather than despairing in the face of multiple diverse models, we can create a composite picture based on a few representative models: we embrace the complexity and thus achieve a richer, more nuanced description. These tools offer both a fluid interaction with the output of the models and a look into their backstage operations.

In sum, this dissertation illustrates why, as descriptive linguists, we shouldn't trust distributional models blindly, but also how we can exploit them nonetheless. On the one hand, it illustrates a workflow for investigating distributional modelling itself: the same steps followed in this study can be

[4]Or of finding ways for other people to do so.

applied to alternative implementations for a better understanding of distributional approaches. On the other hand, with both warnings and suggestions, it offers a framework and tools for future studies implementing token-level distributional models to linguistic research or, as we like to call it, linguistic cloudspotting.

Bibliography

Agirre, Eneko & Philip Edmonds (eds.). 2007. *Word Sense Disambiguation. Algorithms and Applications*. Vol. 33 (Text, Speech, and Language Technology). Springer.

Arppe, Antti, Gaëtanelle Gilquin, Dylan Glynn, Martin Hilpert & Arne Zeschel. 2010. Cognitive Corpus Linguistics: five points of debate on current theory and methodology. *Corpora* 5(1). 1–27.

Barcelona, Antonio. 2015. Metonymy. In Ewa Dabrowska & Dagmar Divjak (eds.), *Handbook of Cognitive Linguistics*, 143–167. Berlin; München; Boston: De Gruyter.

Baroni, Marco, Georgiana Dinu & Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238–247. Baltimore, Maryland: Association for Computational Linguistics.

Baroni, Marco & Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics*, 1–10. Edinburgh, Scotland, UK: Association for Computational Linguistics.

Bolognesi, Marianna. 2020. *Where Words Get their Meaning: Cognitive processing and distributional modelling of word meaning in first and second language* (Converging Evidence in Language and Communication Research). Amsterdam: John Benjamins Publishing Company.

den Boon, Ton, Dirk Geeraerts & Marjan Arts. 2007. *Van Dale klein woordenboek van de Nederlandse taal*. Utrecht: Van Dale.

Bostock, M., V. Ogievetsky & J. Heer. 2011. D^3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17(12). 2301–2309.

Bullinaria, John A. & Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* 39(3). 510–526.

Campello, Ricardo J. G. B., Davoud Moulavi & Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda & Guandong Xu (eds.), *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science), 160–172. Berlin, Heidelberg: Springer.

Card, Stuart K., Jock D. Mackinlay & Ben Shneiderman. 1999. *Readings in information visualization: using vision to think* (The Morgan Kaufmann

Series in Interactive Technologies). San Francisco, Calif: Morgan Kaufmann Publishers.

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert & Barbara Borges. 2021. *shiny: Web Application Framework for R*. R package version 1.6.0. https://shiny.rstudio.com/.

Church, Kenneth Ward & Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *ACL '89: Proceedings of the 27th annual meeting on Association for Computational Linguistic*, 76–83. Association for Computational Linguistics.

Cox, Michael A. A. & Trevor F. Cox. 2008. Multidimensional Scaling. In Chunhouh Chen, Wolfgang Härdle & Antony Unwin (eds.), *Handbook of data visualization* (Springer Handbooks of Computational Statistics), 315–348. Berlin: Springer.

Croft, William. 2003. The role of domains in the interpretation of metaphors and metonymies. In René Dirven & Ralf Pörings (eds.), *Metaphor and metonymy in comparison and contrast* (Mouton Reader), 161–206. Berlin; New York, NY: Mouton de Gruyter.

De Pascale, S. 2019. *Token-based vector space models as semantic control in lexical lectometry*. Leuven: KU Leuven PhD Dissertation.

de Saussure, Ferdinand. 1971. *Cours de linguistique génerale*. Charles Bally & Albert Sechehaye (eds.). Paris: Payot.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*.

de Vries, Wietse, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord & Malvina Nissim. 2019. *BERTje: A Dutch BERT Model*.

Evert, Stefan. 2009. 58. Corpora and collocations. In Anke Lüdeling & Merja Kytö (eds.), *Corpus Linguistics. An International Handbook*, vol. 2 (Handbooks of Linguistics and Communication Science), 1212–1248. Berlin; New York: Mouton de Gruyter.

Firth, John Rupert. 1957. A synopsis of linguistic theory 1930-1955. In John Rupert Firth (ed.), *Studies in Linguistic Analysis* (Special Volume of the Philological Society), 1–32. Oxford: Blackwell.

Fleiss, Joseph L. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76(5). 378–382.

Gablasova, Dana, Vaclav Brezina & Tony McEnery. 2017. Collocations in Corpus-Based Language Learning Research: Identifying, Comparing, and Interpreting the Evidence: Collocations in Corpus-Based Language Learning Research. *Language Learning*.

Gamer, Matthias, Jim Lemon, Ian Fellows & Puspendra Singh. 2019. *Irr: Various Coefficients of Interrater Reliability and Agreement*. manual. https://CRAN.R-project.org/package=irr.

Geeraerts, Dirk. 1988. Where does prototypicality come from? In Brygida Rudzka-Ostyn (ed.), *Current Issues in Linguistic Theory*, vol. 50, 207–229. Amsterdam: John Benjamins Publishing Company.

Geeraerts, Dirk. 1993. Vagueness's puzzles, polysemy's vagaries. *Cognitive Linguistics* 4. 223–272.

Geeraerts, Dirk. 1997. *Diachronic prototype semantics: a contribution to historical lexicology* (Oxford Studies in Lexicography and Lexicology). Oxford; New York: Clarendon Press; Oxford University Press.

Geeraerts, Dirk. 1999. Idealist and empiricist tendencies in Cognitive Linguistics. In Theo Janssen & Gisela Redeker (eds.), *Cognitive Linguistics: Foundations, Scope, and Methodology*, 163–194. Berlin: Mouton de Gruyter.

Geeraerts, Dirk. 2003. The interaction of metaphor and metonymy in composite expressions. In René Dirven & Ralf Pörings (eds.), *Metaphor and metonymy in comparison and contrast* (Mouton Reader), 435–466. Berlin; New York, NY: Mouton de Gruyter.

Geeraerts, Dirk. 2006. *Words and other wonders: papers on lexical and semantic topics* (Cognitive Linguistics Research 33). Berlin; New York: Mouton de Gruyter.

Geeraerts, Dirk. 2010a. The doctor and the semantician. In Dylan Glynn & Kerstin Fischer (eds.), *Quantitative methods in cognitive semantics: corpus-driven approaches* (Cognitive Linguistics Research 46), 63–78. Berlin; New York: De Gruyter Mouton.

Geeraerts, Dirk. 2010b. *Theories of lexical semantics*. Oxford; New York: Oxford University Press. 341pp.

Geeraerts, Dirk. 2015. Sense individuation. In Nick Riemer (ed.), *The Routledge Handbook of Semantics*, 1st edn., 233–247. London: Routledge.

Geeraerts, Dirk. 2016. The sociosemiotic commitment. *Cognitive Linguistics* 27(4). 527–542.

Geeraerts, Dirk. 2017. Distributionalism, old and new. In Anastasiia Makarova, Stephen M. Dickey & Dagmar Divjak (eds.), *Each venture a new beginning: studies in honor of Laura A. Janda*, 29–38. Bloomington, Indiana: Slavica.

Geeraerts, Dirk & Hubert Cuyckens. 2007a. Introducing Cognitive Linguistics. In Dirk Geeraerts & Hubert Cuyckens (eds.), *The Oxford handbook of cognitive linguistics* (Oxford Handbooks), 3–22. Oxford; New York: Oxford University Press.

Geeraerts, Dirk & Hubert Cuyckens (eds.). 2007b. *The Oxford handbook of cognitive linguistics* (Oxford Handbooks). Oxford; New York: Oxford University Press.

Geeraerts, Dirk, Stefan Grondelaers & Peter Bakema. 1994. *The Structure of Lexical Variation: Meaning, Naming, and Context*. Berlin; New York: De Gruyter Mouton.

Geeraerts, Dirk, Stefan Grondelaers & Dirk Speelman. 1999. *Convergentie en divergentie in de Nederlandse woordenschat: een onderzoek naar kledingen voetbaltermen* (Publikaties van het Meertens-Instituut 31). Amsterdam: Meertens-Instituut.

Gibbs, Raymond W., Jr. & Gerard Steen (eds.). 1999. *Metaphor in cognitive linguistics: selected papers from the Fifth International Cognitive Linguistics Conference, Amsterdam, July 1997* (Amsterdam Studies in the Theory and History of Linguistic Science Ser. 4 175). Amsterdam: Benjamins.

Gibbs Jr., Raymond W. (ed.). 2008. *The Cambridge handbook of metaphor and thought*. New York: Cambridge University Press.

Glynn, Dylan. 2010. Corpus-driven Cognitive Semantics. Introduction to the field. In Dylan Glynn & Kerstin Fischer (eds.), *Quantitative methods in cognitive semantics: corpus-driven approaches* (Cognitive Linguistics Research 46), 1–42. Berlin; New York: De Gruyter Mouton.

Glynn, Dylan. 2014. The many uses of *run*: Corpus methods and Socio-Cognitive Semantics. In Dylan Glynn & Justyna A. Robinson (eds.), *Corpus methods for semantics: quantitative studies in polysemy and synonymy* (Human Cognitive Processing volume 43), 117–144. Amsterdam; Philadelphia: John Benjamins Publishing Company.

Glynn, Dylan & Kerstin Fischer (eds.). 2010. *Quantitative methods in cognitive semantics: corpus-driven approaches* (Cognitive Linguistics Research 46). Berlin; New York: De Gruyter Mouton.

Glynn, Dylan & Justyna A. Robinson (eds.). 2014. *Corpus methods for semantics: quantitative studies in polysemy and synonymy* (Human Cognitive Processing volume 43). Amsterdam; Philadelphia: John Benjamins Publishing Company.

Gries, Stefan Th. 2013. 50-something years of work on collocations: What is or should be next… *International Journal of Corpus Linguistics* 18(1). 137–166.

Gries, Stefan Thomas & Anatol Stefanowitsch (eds.). 2006. *Corpora in cognitive linguistics: corpus-based approaches to syntax and lexis* (Trends in Linguistics. Studies and Monographs 172). Berlin; New York: Mouton de Gruyter.

Hahsler, Michael & Matthew Piekenbrock. 2021. *dbscan: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*. R package version 1.1-8. https://github.com/mhahsler/dbscan.

Harris, Zellig S. 1954. Distributional structure. *Word* 10(2-3). 146–162.

Hausser, Jean & Korbinian Strimmer. 2021. *entropy: Estimation of Entropy, Mutual Information and Related Quantities*. R package version 1.3.0. http://www.strimmerlab.org/software/entropy/.

Heylen, Kris, Dirk Speelman & Dirk Geeraerts. 2012. Looking at word meaning. An interactive visualization of Semantic Vector Spaces for Dutch synsets. In *Proceedings of the eacl 2012 Joint Workshop of LINGVIS & UNCLH*, 16–24. Avignon.

Heylen, Kris, Thomas Wielfaert, Dirk Speelman & Dirk Geeraerts. 2015. Monitoring polysemy: Word space models as a tool for large-scale lexical semantic analysis. *Lingua* 157. 153–172.

Hilpert, Martin & David Correia Saavedra. 2017. Using token-based semantic vector spaces for corpus-linguistic analyses: From practical applications to tests of theoretical claims. *Corpus Linguistics and Linguistic Theory* 16(2).

Hilpert, Martin & Susanne Flach. 2020. Disentangling modal meanings with distributional semantics. *Digital Scholarship in the Humanities*.

Ibarretxe-Antuñano, Iraide & Javier Valenzuela (eds.). 2016. *Lingüística cognitiva*. 2da (Autores, textos y temas Filosofía). Barcelona: Anthropos Ed.

Jurafsky, Daniel & James H. Martin. 2020. *Speech and Language Processing*. 3rd edn.

Kaufman, Leonard & Peter J. Rousseeuw. 1990. Partitioning Around Medoids (Program PAM). In *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley Series in Probability and Statistics), 68–125. Hoboken, NJ, USA: John Wiley & Sons, Inc.

Kiela, Douwe & Stephen Clark. 2014. A Systematic Study of Semantic Vector Space Model Parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality*, 21–30. Gothenburg: ACL.

Konopka, Tomasz. 2020. *umap: Uniform Manifold Approximation and Projection*. R package version 0.2.7.0. https://github.com/tkonopka/umap.

Koptjevskaja-Tamm, Maria & Magnus Sahlgren. 2014. Temperature in the word space: Sense exploration of temperature expressions using word-space modelling. In Benedikt Szmrecsanyi & Bernhard Wälchli (eds.), *Aggregating Dialectology, Typology, and Register Analysis*, 231–267. Berlin, Boston: De Gruyter.

Kövecses, Zoltán. 2015. *Where metaphors come from: reconsidering context in metaphor*. New York, NY: Oxford University Press.

Krijthe, Jesse. 2018. *Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation*. R package version 0.15. https://github.com/jkrijthe/Rtsne.

Kristiansen, Gitte, Michel Achard, René Dirven & Francisco José Ruiz de Mendoza Ibáñez (eds.). 2006. *Cognitive linguistics: current applications and future perspectives* (Applications of Cognitive Linguistics 1). Berlin: Mouton de Gruyter.

Kruskal, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1). 1–27.

Lakoff, George & Mark Johnson. 2003. *Metaphors we live by*. Chicago: University of Chicago Press.

Landauer, Thomas K. & Susan T. Dumais. 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.

Langacker, Ronald W. 2008. *Cognitive grammar: a basic introduction*. Oxford; New York: Oxford University Press.

Lapesa, Gabriella & Stefan Evert. 2014. A Large Scale Evaluation of Distributional Semantic Models: Parameters, Interactions and Model Selection. *Transactions of the Association for Computational Linguistics* 2. 531–546.

Lemmens, Maarten. 2015. Cognitive semantics. In Nick Riemer (ed.), *The Routledge Handbook of Semantics*, 1st edn., 90–105. London: Routledge.

Lenci, Alessandro. 2008. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics* 20(1). 1–31.

Lenci, Alessandro. 2018. Distributional Models of Word Meaning. *Annual Review of Linguistics* 4(1). 151–171.

Levy, Omer, Yoav Goldberg & Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics* 3. 211–225.

Maechler, Martin, Peter Rousseeuw, Anja Struyf & Mia Hubert. 2021. *cluster: "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.* R package version 2.1.2. https://svn.r-project.org/R-packages/trunk/cluster/.

McEnery, Tony & Andrew Hardie. 2012. *Corpus linguistics: method, theory and practice* (Cambridge Textbooks in Linguistics). Cambridge; New York: Cambridge University Press.

McEnery, Tony, Richard Xiao & Yukio Tono. 2010. *Corpus-based language studies: an advanced resource book*. Reprinted (Routledge Applied Linguistics). London: Routledge.

McInnes, Leland, John Healy & Steve Astels. 2016. *How HDBSCAN Works — Hdbscan 0.8.1 Documentation*. https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html (30 April, 2021).

McInnes, Leland, John Healy & James Melville. 2020. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*.

McWhite, Claire D. & Claus O. Wilke. 2020. *colorblindr: Simulate colorblindness in R figures*. R package version 0.1.0. https://github.com/clauswilke/colorblindr.

Mikolov, Tomas, Kai Chen, Greg Corrado & Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space*.

Montes, Mariana & Kris Heylen. Submitted. Visualizing Distributional Semantics. In Dennis Tay & Molly Xie Pan (eds.). Mouton De Gruyter. Submitted.

Montes, Mariana & QLVL. 2021. *QLVL/NephoVis: Altostratus*. Version v1.0.0. Zenodo. https://doi.org/10.5281/ZENODO.5116843.

Okabe, Masataka & Kei Ito. 2002. *Color Universal Design (CUD). How to Make Figures and Presentations That Are Friendly to Colorblind People*. J*Fly Data Depository for Drosophila researchers. https://jfly.uni-koeln.de/color/ (13 July, 2021).

Oksanen, Jari, F. Guillaume Blanchet, Michael Friendly, Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, Eduard Szoecs & Helene Wagner. 2020. *vegan: Community Ecology Package*. R package version 2.5-7. https://CRAN.R-project.org/package=vegan.

Ordelman, Roeland J.F., Franciska M.G. de Jong, Adrianus J. van Hessen & G.H.W. Hondorp. 2007. TwNC: a multifaceted dutch news corpus. *ELRA Newsletter* 12(3-4).

Oskolkov, Nikolay. 2021. *How Exactly UMAP Works*. Medium. https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668 (7 May, 2021).

Pantel, Patrick & Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02*, 613. Edmonton, Alberta, Canada: ACM Press.

Perek, Florent. 2016. Using distributional semantics to study syntactic productivity in diachrony: A case study. *Linguistics* 54(1). 149–188.

Perek, Florent. 2018. Recent change in the productivity and schematicity of the way-construction: A distributional semantic analysis. *Corpus Linguistics and Linguistic Theory* 14(1). 65–97.

Raganato, Alessandro, Jose Camacho-Collados & Roberto Navigli. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 99–110. Valencia, Spain: Association for Computational Linguistics.

Rohrer, Tim. 2007. Embodiment and experientialism. In Dirk Geeraerts & H. Cuyckens (eds.), *The Oxford handbook of cognitive linguistics* (Oxford Handbooks), 25–47. Oxford; New York: Oxford University Press.

Rosch, Eleanor. 1978. Principles of categorization. In Eleanor Rosch & Barbara B. Lloyd (eds.), *Cognition and Categorization*, 27–48. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Rousseeuw, Peter J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20. 53–65.

Rudzka-Ostyn, Brygida (ed.). 1988. *Topics in Cognitive Linguistics*. Vol. 50 (Current Issues in Linguistic Theory). Amsterdam: John Benjamins Publishing Company.

Sahlgren, Magnus. 2006. *The word-space model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Stockholm: Dep. of Linguistics, Stockholm Univ. [u.a.]

Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Linguistics* 20(1). 33–53.

Schütze, Hinrich. 1998. Automatic Word Sense Discrimination. *Computational Linguistics* 24(1). 97–123.

Semino, Elena. 2008. *Metaphor in discourse*. Cambridge, UK; New York: Cambridge University Press.

Shneiderman, Ben. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *IEEE Visual Languages*, 96–13.

Sievert, Carson, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec & Pedro Despouy. 2021. *plotly: Create Interactive Web Graphics via plotly.js*. R package version 4.9.4.1. https://CRAN.R-project.org/package=plotly.

Sinclair, John. 1991. *Corpus, concordance, collocation*. 3. impr (Describing English Language). Oxford: Oxford Univ. Press.

Sinclair, John. 1998. The Lexical Item. In Edda Weigand (ed.), *Contrastive lexical semantics* (Amsterdam Studies in the Theory and History of Linguistic Science Series 4, Current Issues in Linguistic Theory 171), 1–24. Amsterdam: Benjamins.

Smilkov, Daniel, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. Viégas & Martin Wattenberg. 2016. Embedding Projector: Interactive Visualization and Interpretation of Embeddings. In arXiv:1611.05469.

Speelman, Dirk & Kris Heylen. 2014. *semvar: Semantic Variation*. R package version 0.1.1.

Speelman, Dirk & Kris Heylen. 2017. From dialectometry to semantics. In Martijn Wieling, Gosse Bouma & Gertjan van Noord (eds.), *From Semantics to Dialectometry (Festschrift John Nerbonne)*, 325–334. Groningen: University of Groningen.

Stefanowitsch, Anatol. 2010. Empirical cognitive semantics: Some thoughts. In Dylan Glynn & Kerstin Fischer (eds.), *Quantitative methods in cognitive semantics: corpus-driven approaches* (Cognitive Linguistics Research 46), 355–380. Berlin; New York: De Gruyter Mouton.

van Sterkenburg, Piet (ed.). 1991. *Van Dale groot woordenboek van hedendaags Nederlands*. 2. dr (Van Dale Woordenboeken Voor Hedendaags Taalgebruik). Utrecht: van @Dale Lexicografie.

Stubbs, Michael. 2009. Memorial Article: John Sinclair (1933–2007). *Applied Linguistics* 30(1). 115–137.

Tognini-Bonelli, Elena. 2001. *Corpus linguistics at work*. Amsterdam; Philadelphia: J. Benjamins.

Turney, Peter D & Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research* 37. 141–188.

Van de Cruys, Tim, Thierry Poibeau & Anna Korhonen. 2013. A Tensor-based Factorization Model of Semantic Compositionality. In *Proceedings of NAACL 2013*, 1142–1151. Atlanta, Georgia, USA.

van der Maaten, L.J.P. 2014. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research* 15. 3221–3245.

van der Maaten, L.J.P. & G.E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research* 9. 2579–2605.

van Noord, Gertjan. 2006. At Last Parsing Is Now Operational. In *Actes de la 13ème conférence sur le Traitement Automatique des Langues Naturelles. Conférences invitées*, 20–42. Leuven, Belgium: ATALA.

Wattenberg, Martin, Fernanda Viégas & Ian Johnson. 2016. How to Use t-SNE Effectively. *Distill* 1(10). e2.

Wielfaert, Thomas, Kris Heylen, Dirk Speelman & Dirk Geeraerts. 2019. Visual Analytics for Parameter Tuning of Semantic Vector Space Models. In Miriam Butt, Annette Hautli-Janisz & Verena Lyding (eds.), *LingVis: visual analytics for linguistics* (CSLI Lecture Notes no. 220), 215–245. Stanford, California: CSLI Publications, Center for the Study of Language and Information.

Wittgenstein, Ludwig. 1958. *Philosophical investigations*. Trans. by G. E. M. Anscombe. 2nd ed., repr. Cambridge, Mass: Blackwell.

Yarowsky, David. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, 189–196. Cambridge, Massachusetts, USA: Association for Computational Linguistics.