Modelling meaning granularity of nouns with vector space models

Mariana Montes

KU Leuven

The present study is part of a larger research project with the goal of developing tools for large-scale corpus-based semantic analyses. One such tool involves representing semantic structure with vector space models (VSMs), which currently requires a deeper understanding of its inner workings and how its results relate to cognitive theories of meaning. As a distributional method, it builds on the context of a lexical item to describe it and compare it to others, which raises our main research question: how is that context defined, and how does it vary for different semantic phenomena? We investigate how models based on different parameter settings deal with a range of semantic issues, such as granularity of meaning. For this purpose, a set of 7 polysemous homonyms in Dutch was selected to test how the distance between homonymous usages of a lexical item are represented in relation to polysemous usages within the same homonym. The models were built from a 520MW corpus of contemporary Dutch and Flemish newspapers and the resulting VSMs were evaluated through visual analytics, via scatterplots where more similar tokens appear closer to each other. In addition, colorcoding of manual tags allows us to compare how they were grouped by human annotators and by the computational models in a way that is consistent with the cognitive approach to meaning and categorization. The results indicate that not one set of parameters deals well with granularity in all cases. For example, those that disambiguate *stof* ‘substance/fabric/topic, dust’ well fail with *hoop* ‘heap, hope’ and vice versa. Furthermore, for some nouns some senses may be well grouped while the homonyms are not.

1. Introduction

Usage-based linguistics has a lot to gain from Big Data. The availability of large amounts of textual data and computational tools to process it constitutes an attractive source for empirical analysis of language in use. However, understanding of the tools and methods necessary to take advantage of these resources doesn't always come hand in hand with understanding of and interest in linguistic issues.

This paper presents a case study within a larger project aimed to merging these branches: applying tools from computational linguistics to research in lexical semantics. One such tool involves representing semantic structure with vector space models (VSMs), an established computational technique (see Turney & Pantel 2010 for an overview) that still requires a deeper understanding of its inner workings and how its results relate to cognitive theories of meaning.

Count-based VSMs represent words as vectors of co-occurrence frequencies in a multidimensional space (Lenci 2018). As a distributional method, VSMs build on the context of a lexical item to describe it and compare it to others, which raises our main research question: how is that context defined, and how does it vary for different semantic phenomena, where the various context words play different roles? Although token-based VSMs are increasingly used in corpus-based cognitive semantics, we believe it is insufficiently appreciated how alternative parameter settings impact the final output. Accordingly, we investigate how models based on different parameter settings deal with a range of semantic issues, such as, in the case study described here, granularity of meaning. For that purpose we selected 7 Dutch nouns that present both homonymy and polysemy: each form has at least two very distinct, unrelated meanings, i.e. homonyms, and at least one of them presents polysemous (distinct but semantically related) usages[1]. The goal was to test how the distance between homonymous usages of a lexical item is represented in relation to polysemous usages within the same homonym. Ideally, tokens belonging to different homonyms will be far away from each other in the vector space and form two distinct groups, while the different sense distinctions within a homonym will be harder to pull apart. For example, literal and metaphorical senses of Dutch *hoop* 'heap' should be closer together and far from its homonym meaning 'hope'.

We will describe the technique in Section 2 and specify its application for the case study in Section 3. Results for the specific cases of Dutch *hoop* 'hope, heap' and *stof* 'substance/fabric/topic, dust' will be presented in Section 4 and we will finish with a conclusion in Section 5.

2. Distributional semantics

It this study we use vector space models (VSMs), a computational technique (Lenci 2018, Turney & Pantel 2010), to model the semasiological structure of lexical

[1] The forms are *blik* 'view, tin', *hoop* 'hope, heap', *horde* 'horde, hurdle', *schaal* 'scale, dish', *spot* 'ridicule, spot(light)', *staal* 'steel, sample' and *stof* 'substance/fabric/topic, dust'. Sense tags were based on *Van Dale* (Sterkenburg, 1991) entries.

items. The method can take multiple forms and, while it constitutes a promising tool for exploiting the current Big Data availability in lexicological research, it is not so popular outside computational linguistics. A key distinction between the approach followed here and other computational perspectives on sense disambiguation (e.g. Reisinger & Mooney 2010, Neelakantan *et al.* 2015) is the goal and, therefore, the evaluation mechanism. Rather than aiming for a more precise result in NLP tasks, our ultimate goal is to aid lexicographical research with computational models. As such, we are interested in figuring out the linguistic interpretation of the results, if and how computational models agree with manual annotations, and the relationship between parameter settings (i.e. a model's way of defining and representing context) and output.

In this section we will offer a brief overview of what is known as VSMs and how they have been used in the area of cognitive linguistics. We will describe count-based models, as opposed to prediction-based models, which use neural networks. We use the former rather than the latter because the connection between the parameter settings and the final output is more transparent. Nevertheless, they are indeed an interesting option to expand this line of research —especially architectures such as BERT (Devlin *et al.* 2019), since they can represent individual tokens as well.

2.1. *Vector space models*

At the core of vector space models, *aka* distributional models, we find the Distributional Hypothesis, which is most often linked to Harris's observation that "difference of meaning correlates with difference of distribution" (1954:156). In other words, items that occur in similar contexts in a given corpus will be semantically similar, while those that occur in different contexts will be semantically different. Distributional models operationalize this idea by representing words as vectors (i.e. arrays of numbers) coding frequency information. Typically, the raw frequency is transformed to some association strength measure, such as pointwise mutual information (PMI, see Church & Hanks 1989), which compares the frequency with which two words occur close to each other and the expected frequency if the words were independent. Since negative PMI values tend to be unreliable (Bullinaria & Levy 2007, Kiela & Clark 2014, Jurafsky & Martin 2020:109), PPMI (positive PMI) is used instead, by turning the negative PMI values to zeros. For example, Table 1[2] shows small vectors representing the English nouns *linguistic*, *lexicography*, *research* and *chocolate*, as well as the adjective *computational*, as series of association strengths with a set of lemmas.

Table 1: Example of type-level vectors.

target	language/n	word/n	flemish/j	english/j	speak/v
linguistics/n	4.37	0.99	0	3.16	0.41
lexicography/n	3.51	2.18	0	2.19	2.09
computational/j	1.60	0.08	0	0	0
research/n	0.20	0	0.04	0	0
chocolate/n	0	0	1.28	0	0

Each row is a vector coding the distributional information of the lemma it represents. As we can see in this example, words with similar vectors (e.g. *linguistics* and *lexicography*) are semantically similar, while words with different vectors (e.g. *linguistics* and *chocolate*) are semantically different.
The vectors in Table 1 are type-level vectors: each of them aggregates over all the instances of a given word, e.g. *linguistics*, to build an overall profile. As a result, it collapses the internal variation of the lemma, i.e. its semasiological structure. One way of uncovering such information is 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 occurrences of *study* in (1) through (3), where the target item is 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 values as the individual window size comprises,

[2] PPMI values in these table are based on symmetric window of 10 in the GloWbE corpus. The letter to the right of the word indicates the part-of-speech: *n* for nouns, *j* for adjectives and *v* for verbs.

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), we replace the context words around the token with their respective type-level vectors (De Pascale 2019, Heylen *et al*. 2015). For example, we could represent example (1) with the vector for its context word *lexicography*, that is, the second row in Table 1; example (2) with the sum of the vectors for *linguistics* (row 1) and *computational* (row 3); and example (3) with 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*.

From applying this method we obtain numerical representations of occurrences of a word. We can compare them to each other by calculating pairwise distances, which is at the base of clustering analyses and visualization techniques based on dimensionality reduction.

2.2. *Distributional models in cognitive linguistics*

Vector space models were originally developed in computational linguistics but they are increasingly being used in corpus-based cognitive linguistics, particularly in lexical semantics (Heylen *et al*. 2012, 2015), diachronic construction grammar (Hilpert & Correia Saavedra 2017, Hilpert & Flach 2020, Perek 2016, 2018), lectometry (De Pascale 2019, Ruette *et al*. 2016), and lexical typology (Koptjevskaja-Tamm & Sahlgren 2014). Most of them make use of type-level vectors, while some include token-level vectors as well.

These developments notwithstanding, we believe it is insufficiently appreciated how alternative parameter settings impact the final representation, at either token or type level. A typical approach involves relying on previous work and overview papers such as Kiela and Clark's (2014) study (see also Baroni *et al*. 2014). Such overview studies explore a large parameter space, i.e. a variety of choices regarding multiple parameter settings for the models, and compare their performance in terms of accuracy in relation to a benchmark.

However, this method does not tell us how the models agree with each other, i.e. whether their disagreement with the benchmark pertains to the same cases (Heylen *et al*. 2015:161). Moreover, it considers the manual annotation as a categorical ground truth to aim for, when we know that discrete categories such as these are abstractions (Glynn 2014) and can be restrictive, covering one of multiple possible dimensions of meaning. The approach followed here, instead, uses manual annotation as a heuristic but does not consider neither the dictionary-based definitions nor the models' output as a unique, ultimate description of semasiological structure.

3. Methods

In order to examine how these VSMs deal with meaning granularity, a set of 7 polysemous homonyms in Dutch was selected. Homonyms are semantically more different from each other than senses within a homonym; therefore, based on the Distributional Hypothesis, we would expect that the models would more easily discriminate the former than the latter.

For the purposes of this article we will focus on *hoop* 'hope, heap' and *stof* 'substance/fabric/topic, dust', which were annotated with the tags shown in Tables 2 and 3 respectively.

Table 2: Sense tags for *hoop* their frequencies in the annotated sample.

Tag	Definition	Dutch example	Translation	Freq.
hoop_1	1.1 unordered mass	*een hoop rommel, gooi maar op de hoop*	a pile of junk, you may drop it on the pile	17
hoop_2	1.2 great quantity	*een hoop mensen, een hele hoop geld*	a bunch of people, a lot of money	59
hoop_3	2 positive expectation, trust in something positive	*hoop koesteren, de hoop uitspreken dat...*	to have hope, express the hope that...	236

Table 3: Sense tags for *stof* their frequencies in the annotated sample.

Tag	Definition	Dutch example	Translation	Freq.
stof_1	1.1 matter, substance of a certain kind	*giftige stoffen*	poisonous substances	145

Tag	Definition	Dutch example	Translation	Freq.
stof_2	1.2 fabrics	*wollen en katoenen stoffen*	woolen and cotton fabrics	54
stof_3	1.3 topic about which people talk, write, think, etc.	*stof voor een roman, stof tot onenigheid*	material for a novel, topic of disagreement	37
stof_4	2.1 mass of very small dry particles of various origin, floating in the air	*een wolk stof, stof afnemen*	a cloud of dust, to dust	39
stof_6	2.3 idiomatic uses of "dust"	*stof doen opwaaien*	stir up dust	38

Nonconsecutive numbers in the sense tags correspond to not attested senses from our original selection.

For each of the lemmas we extracted 240-320 tokens and built 212 models by combining different parameter settings, including, non-exhaustively: window size and part-of-speech filter for bag-of-words models, templates based on dependency information, and PPMI weighting. The frequency information for the vectors and the concordances come from a 520MW corpus of contemporary (1999-2004) Dutch and Flemish newspapers (see De Pascale 2019:30), with PPMI values based on a symmetric window of 4 slots to each side of the target. We considered all lemmas with a minimum relative frequency of 1 in 2 million[3], excluding punctuation. At all levels, targets and context words were defined by a combination of stem and part-of-speech: *hoop/noun* refers to the noun ‘hope, heap’, while *hoop/verb* would refer to the verb ‘to hope’, and these are taken to be different items. We compared vectors by using the cosine distance but then applied a log transformation over the ranks to enhance the weight of the smallest distances,

[3] This threshold seemed reasonable given the size of the corpus, but it was a practical choice more than a principled one. Whether the results would be better with a higher threshold is an empirical question that was not addressed.

i.e. the largest similarities. We also applied a range of visualization techniques, but for this article we'll show the t-SNE representations with perplexity 30 (Krijthe 2015, Maaten & Hinton 2008). The goal of this technique is to project the distances between items, originally based on hundreds or thousands of dimensions, to 2 dimensions that we can interpret on the screen. Naturally, this results in loss of information (Wattenberg *et al*. 2016), but it also makes it possible to find patterns in the relative distances that we cannot access otherwise.

Finally, we applied partitioning around medoids[4] to select eight representative models that can give us an idea of the variation among them. They were visualized with an interactive Javascript tool[5] that, among other things, allows the user to select portions of the plots and identify which context words characterize the selected tokens across the different models (Montes & Wielfaert 2021).

4. Results

Both nouns, *hoop* and *stof*, are homonymous and polysemous. If semantic similarity can be operationalized as distributional similarity and granularity of meaning can be mapped to relative distances in the modelled space, we would expect a clear distinction between tokens of different homonyms, and a less clear distinction between tokens belonging to different senses of the same homonym. In the visualization, this would be translated as distinct areas for each homonym and probably more overlap between the senses of one homonym. However, this is surprisingly hard to achieve: the different degrees of granularity don't seem to have a mapping to the visual representation of the models.

4.1. *Plotting the "best" models*

Figure 1 shows two medoids (i.e. representative models) for each of the lemmas. The top row shows the "best" medoid of each lemma if we consider manual annotation as a benchmark, i.e. the model that best separates the tokens of different senses. However, the parameter settings that work best for one lemma do not perform well for the other: in the second row we show the result of this switch. In other words, panels (a) and (b) show the best models for *hoop* and *stof* respectively, panel (c) shows the effect of applying the parameters from (b) to *hoop*, and panel (d) the effect of applying the parameters from (a) to *stof*.

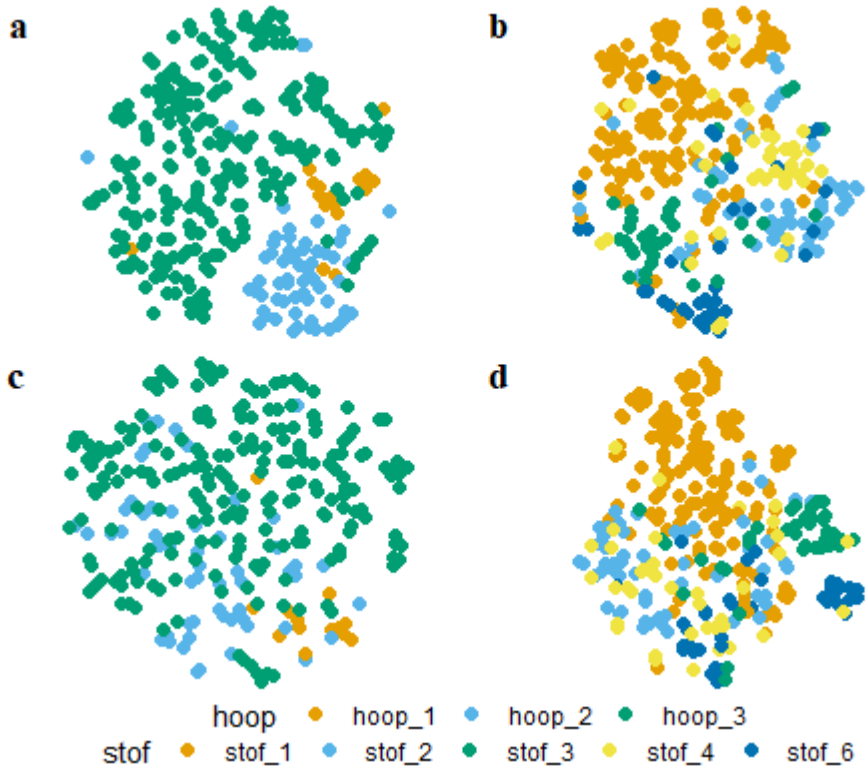
The parameters that result in these models are in fact very different, although their second-order configuration is equivalent: the dimensions of the type-level vectors representing the context words of the target are the union of the first-order context

[4] Run with the pam() function of the {cluster} R package (Maechler *et al*. 2019).

[5] For other visualization techniques, the full range of models and other lemmas the reader may explore https://qlvl.github.io/NephoVis/.

words collected throughout the corresponding sample. As a result, the dimensionality of the token-level vectors is already quite low: 833 for (a), 483 for (b), 352 for (c), and 849 for (d)[6].

For panels (a) and (d) we have a dependency-based model that takes the context words linked up to three steps away from the target in the dependency path, and weights the contribution of each item on that distance. This is illustrated by the superscripts in examples (4) and (5).



[6] The fact that the *number* of dimensions in each model is different is irrelevant in this comparison: the parameter setting simply requires that the dimensions of the type-level vectors are the same features identified in the context of the tokens. In practice, the result is not very different from selecting the 5000 most frequent lemmas as dimensions instead.

Figure 1: (a) Best model of *hoop*; (b) Best model of *stof*; (c) Model of *hoop* with best parameters of *stof*; (d) Model of *stof* with best parameters of *hoop*.

In (4), the determiner *een* ‘a’ 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 (5) offers a much more complex picture, particularly because the link between the target *hoop* ‘hope’ and the core of the sentence, the verb *uitspreken* ‘to express’ (split in *sprak* and its particle *uit*), is short and opens the path to many other elements in the sentence.

(4) Er^{3} is^{2} een^{1} **hoop** onzin^{1}, talent is^{3} niet iedereen gegeven.
There is a ***lot of*** *nonsense, talent is not given to everyone.*
(*Algemeen Dagblad*, 27/10/2001)

(5) 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.
The trainer of FC Utrecht also expressed the ***hope*** *that the club management would allow him to invest again soon.*
(*NRC Handelsblad*, 24/05/2004)

A key point for this lemma is that *hoop* ‘hope’ is a mass noun, and therefore often occurs with the definite determiner *de* (40% of the cases), whereas *hoop* ‘heap’ tends to occur with *een* (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 regardless of their relationship to the target, thus drowning this pattern in noise.
In contrast, the parameter settings that result in the plots in panels (b) and (c) 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 PPMI with the target lemma is larger than 0. In the case of (6), for example, the model selects *discussie* ‘discussion’ and *oplevert* ‘yields’. Words that might follow after the period and those before *film* ‘movie’ are excluded by this model. 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* ‘conclusion’, which survive the window size and part-of-speech filters, are excluded by the association strength filter.

(6) Dit is een perfect voorbeeld van een film die na afloop veel **stof** tot *discussie oplevert*.
*This is a perfect example of a film that afterwards leaves a lot [of **stuff**] to discuss.*
(*Algemeen Dagblad*, 11/12/2003)

For each lemma we have 8 representative models covering the variation across the 212 generated models. The models shown in Figure 1(a) and 1(b) are, for each of these lemmas, the representatives that best match the sense annotation, but we can quite confidently assume that the rest of the models they represent are also better matches than models in other clusters. The “best” *hoop* medoid represents 23 models selecting context words based on the length of the dependency path (up to 3 slots away, with or without weight) but not weighting them with PPMI. In contrast, the “best” *stof* medoid represents 30 models sharing the following characteristics:

- Bag-of-words model with window larger than 3 and part-of-speech filter.
- Second order dimensions are also filtered by part-of-speech (nouns, adjectives and verbs).
- The tokens are not weighted by PPMI, unless the first order window is 5-5.

4.2. *Granularity of meaning*

The expectation of a visual representation of granularity of meaning is fulfilled for *hoop* in Figure 1(a): there is a greater distinction between the homonyms than between the senses of the polysemous homonym. In contrast, even the best model of *stof* presents a structure that does not match the hierarchy of homonymy and polysemy. First, we tend to find an isolated group of tokens corresponding to the idiomatic usages of ‘dust’ (*stof_6*, as shown in (7)), far from the tokens of the same homonym (literal uses of ‘dust’).

(7) Het debuut van regisseur Nabil Ayouch deed in zijn thuisland veel *stof* opwaaien.
*Director Nabil Ayouch’s debut stirred up a lot of **dust** in his homeland.*
(*Volkskrant*, 06/01/2000)

Second, the most frequent sense, *stof_1* ‘substance’, exhibits internal groups characterized by the co-occurrence of very frequent collocates, such as *gevaarlijk* ‘dangerous’ and *schadelijk* ‘damaging’, as shown in (8).

(8) Vooral bij westenwind vrezen de bewoners lawaai, stank en schadelijke *stoffen*.
Especially with the western wind, the inhabitants fear noise, stench and damaging ***substances****.*
(*NRC Handelsblad*, 10/09/2001)

Third, *stof_2* ‘fabric’ and *stof_4* ‘dust’ stick together in many representative models although they are senses of different homonyms. This can be explained by the fact that both senses tend to co-occur with quite concrete context words, such as names for materials and colors, while the ‘substance’ sense shown in (8) occurs in more chemical-related contexts and the ‘topic’ sense illustrated in (6) co-occurs with the semantic domain of communication instead. In other words, this is not necessarily a failure of the models nor an inherent inadequacy of the lexicographical model, but rather a mismatch between the two perspectives, between distributional structure and semantic structure.

5. Conclusion

The goal of the research illustrated in this paper was to understand the relationship between parameter settings and output of VSMs in order to guide lexicographical applications. However, the results shown here suggest that there is no set of parameters that works best across the board. This is made clear by the comparison between the models that best match the sense annotation of two lemmas exhibiting both homonymy and polysemy and by switching the corresponding parameter settings: settings that succeed in one lemma fail in the other. Instead, the output is crucially sensitive to the particular distributional behavior of each lemma. On the one hand, the collocational patterns identified by the models do not necessarily correspond to the phenomenon we want to model. In the case of *stof,* specific collocates characterize different senses to a certain degree, but do not discriminate between homonyms that well. On the other hand, even if there are semantically distinctive patterns to be found, they are not identified by the same parameter settings: a syntactically informed model is required to identify determiners as a discriminating features of the *hoop* homonyms, while the same information proves less useful in regard to *stof*.

We make no claims regarding the psychological validity of either VSMs or manual disambiguation. Rather than championing one or the other, we suggest a combined approach that enriches lexicographical description with output from VSMs instead of replacing it. In addition, we would discourage a selection of parameters based on overviews that measure accuracy across multiple words based on benchmarks that might not match the interest of the lexicographer to begin with. Instead, we would recommend the exploration of several representative models understood as multiple representations of *distributional* structure that need not match the lexicographer's initial *semantic* categories.

References

Baroni, M., Dinu, G., and Kruszewski, G. (2014). 'Don't count, predict! A systematic comparison of context-counting vs. Context-predicting semantic vectors'. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 238–247.

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

Church, K. W., and Hanks, P. (1989). 'Word association norms, mutual information, and lexicography', In *ACL '89: Proceedings of the 27th annual meeting on Association for Computational Linguistics*, 76–83.

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

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. *arXiv:1810.04805 [Cs]*.

Glynn, D. (2014). 'The many uses of *run*: Corpus methods and Socio-Cognitive Semantics'. In D. Glynn and J. A. Robinson, eds., *Corpus methods for semantics: Quantitative studies in polysemy and synonymy*, Amsterdam; Philadelphia: John Benjamins Publishing Company, 117–144.

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

Heylen, K., Speelman, D., and Geeraerts, D. (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*, Avignon, 16–24.

Heylen, K., Wielfaert, T., Speelman, D., and Geeraerts, D. (2015). 'Monitoring polysemy: Word space models as a tool for large-scale lexical semantic analysis'. *Lingua*, 157, 153–172.

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

Hilpert, M., and Flach, S. (2020). 'Disentangling modal meanings with distributional semantics'. *Digital Scholarship in the Humanities*.

Jurafsky, D., and Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.).

Kiela, D., and Clark, S. (2014). ‘A Systematic Study of Semantic Vector Space Model Parameters’. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality*, Gothenburg: ACL, 21–30.

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

Krijthe, J. H. (2015). *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*. Retrieved from https://github.com/jkrijthe/Rtsne

Lenci, A. (2018). ‘Distributional Models of Word Meaning’. *Annual Review of Linguistics*, 4(1), 151–171.

Maaten, L. van der, and Hinton, G. (2008). ‘Visualizing Data using t-SNE’. *Journal of Machine Learning Research*, 9, 2579–2605.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2019). *Cluster: Cluster Analysis Basics and Extensions*.

Montes, M., and Wielfaert, T. (2021). ‘QLVL/NephoVis: Altostratus', Zenodo. doi:10.5281/ZENODO.5116843

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2015). ‘Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space’. *ArXiv:1504.06654 [Cs, Stat]*.

Perek, F. (2016). ‘Using distributional semantics to study syntactic productivity in diachrony: A case study’. *Linguistics*, 54(1), 149–188.

Perek, F. (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.

Reisinger, J., and Mooney, R. J. (2010). ‘Multi-Prototype Vector-Space Models of Word Meaning’. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 109–117. https://aclanthology.org/N10-1013

Ruette, T., Ehret, K., and Szmrecsanyi, B. (2016). ‘A lectometric analysis of aggregated lexical variation in written Standard English with Semantic Vector Space models’. *International Journal of Corpus Linguistics*, 21(1), 48–79.

Schütze, H. (1998). ‘Automatic Word Sense Discrimination’. *Computational Linguistics*, 24(1), 97–123.

Sterkenburg, P. van (Ed.). (1991). *Van Dale groot woordenboek van hedendaags Nederlands*. van @Dale Lexicografie.

Turney, P. D., and Pantel, P. (2010). ‘From Frequency to Meaning: Vector Space Models of Semantics’. *Journal of Artificial Intelligence Research*, 37, 141–188.

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to Use t-SNE Effectively. *Distill*, *1*(10), e2.