Chapter 33

From dialectometry to semantics

Dirk Speelman
University of Leuven

Kris Heylen
University of Leuven

Aggregate-level studies of linguistic variation typically adopt an onomasiological perspective on linguistic variation. Recently, however, a number of corpus-based techniques have been developed in the distributional semantics framework to detect semantic shifts across large corpora (Sagi, Kaufmann & Clark 2011; Cook & Hirst 2011; Gulordava & Baroni 2011). In this chapter, we apply one such technique to the corpus-based, aggregate-level investigation of semasiological regional variation in Dutch. More specifically, we use token-based vector space models, in which (a random subset of) the tokens of a target word are represented as a 'token cloud in vector space'. In order to compare the use of a target word in two regional varieties of Dutch, viz. Netherlandic Dutch and Belgian Dutch, we build a token cloud for each variety and superimpose both token clouds. Next, we apply measures, which we call *separation indices*, and which quantify to which extent the superimposed clouds exhibit non-overlapping areas (or areas with less overlap). Such areas are of interest because they signal possible differences in the (number of) senses or usage patterns of the word in both varieties. The purpose of these *separation indices* is to quantify semasiological distances between language varieties and by consequence to allow for a (dia)lectometric approach to the study of semasiological variation. This chapter reports on a methodological pilot study that investigates the merits of four candidate types of *separation indices*.

1 Introduction

The methods that were developed within the framework of dialectometry (Nerbonne & Kretzschmar 2003) have been a rich source of inspiration for many different types of studies into aggregate-level language variation, both within dialectometry *sensu stricto* and in lectometry more in general — we use 'lectometry' as an umbrella term for dialectometry, stylometry, sociolectometry, etc. In this chapter, we specifically zoom in on corpus-based studies of regional and register variation that adopt

a 'lectometric approach'. A specific challenge in all corpus-based lectometric studies on lexical/grammatical variation is dealing with semantic differences. Whereas in survey data of the type often used in dialectometric studies, the context in which words/expressions are used by survey participants is kept constant, such consistency does not apply to corpus materials. The contexts in which words are used in corpora vary dramatically from one instance to another and issues related to e.g. polysemy and vagueness are hard to address, especially when the corpora are large and manual inspection of all usage instances is not an option.

Semantic vector space models, or simply vector space models or VSMs (Turney & Pantel 2010), a technique that is often used in natural language processing in tasks such as thesaurus extraction and word sense disambiguation, offer promising possibilities for semi-automatically accounting for lack of synonymy in corpus-based studies of lexical/grammatical variation. For instance, Ruette et al. (2014) incorporate VSMs into a 'lectometric framework'. In that study, in which an onomasiological perspective is adopted, a weighting mechanism is installed that penalizes, in the 'lectometric calculations', data points that occupy a peripheral position according to the semantic similarity scores that were derived from VSMs. This way the effect of potentially problematic data points on the lectometric results is reduced.

In this chapter, rather than attempting to neutralize potential noise coming from non-synonymy, we make non-synonymy, or rather, semasiological variability, the topic of (lectometric) investigation, thus switching to a semasiological perspective. VSMs play a crucial role in our approach. Using token-based VSMs, we build so-called token clouds, for a specific target word, for two regional varieties of Dutch, viz. Netherlandic Dutch and Belgian Dutch and we superimpose both token clouds.

Token clouds will be explained in more detail in Section 2. For the time being, Figure 1 informally illustrates the concept. All panels in Figure 1 show token clouds for the Dutch word *monitor*. In the middle panel we see a token cloud for Netherlandic Dutch (NL), with individual points representing individual tokens. Proximity between tokens is a proxy for semantic similarity of the usage contexts of the tokens. In the right panel we see a token cloud for Belgian Dutch (BE). In the left panel both clouds are superimposed. The example illustrates that there is an area where there are only Belgian Dutch tokens (roughly coinciding with the bottom right quadrant of the plots). Manual inspection of the tokens reveals that by and large the tokens in this area are tokens where *monitor* has the meaning YOUTH LEADER, whereas in the other parts of the plot, where the token clouds do overlap, most tokens have either the meaning COMPUTER SCREEN or the meaning SCREEN OF A MEDICAL DEVICE. As it turns out, in Netherlandic Dutch the word *monitor* lacks the meaning YOUTH LEADER, which is exactly what is reflected in the presence of the non-overlapping area.

Having built our token clouds, we then apply measures, which we call *separation indices*, and which quantify to which extent the superimposed clouds exhibit non-overlapping areas (or areas with less overlap). As illustrated in the example in Figure 1, such areas are of interest because they signal possible differences in the (number of) senses or usage patterns of the word in both varieties. The purpose of

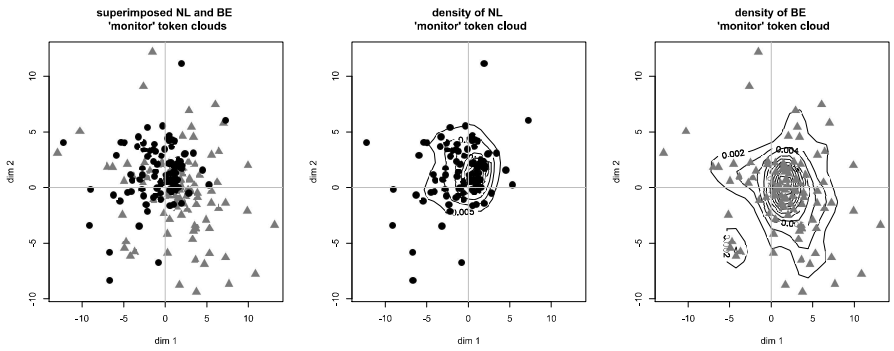

Figure 1: Example of two superimposed token clouds.

these *separation indices* is to quantify semasiological distances between language varieties and by consequence to allow for a (dia)lectometric approach to the study of semasiologic variation. This chapter reports on a pilot study that investigates the merits of four candidate types of *separation indices*.

The structure of the chapter is as follows. In Section 2, we explain how the token clouds are built and we describe four candidate types of *separation indices*. In Section 3, we present results from a case study in which we explore to which extent these *separation indices* yield sensible results. In Section 4, finally, we draw some conclusions.

2 The method: token clouds and separation indices

In this section, we first explain how we build token clouds in vector space. Next, we describe the different candidate types of *separation indices* that will be used in the case study in the next section.

2.1 Building VSMs

The most commonly used kind of VSMs are so-called type-based VSMs. These VSMs are matrices in which the usage, in the corpus, of each target word is summarized in a single row. Columns represent so-called features. In the VSMs used in this chapter, these features simply are words that occur in the vicinity of the target word. The cells in the matrix express the frequency with which each target word co-occurs with each feature, or rather, they contain so-called positive pointwise mutual information values (PPMI) that are derived from the raw co-occurrence frequencies. PPMI values express the 'attraction' between a target word and a feature.

The rationale then is that, given a large enough corpus, target words with similar meanings tend to have similar row vectors. Therefore, distances between row vectors (typically calculated as one minus the cosine of the angle between the two row vec-

tors) can be used as a proxy for differences in meaning/usage. Table 1 illustrates the structure of a type-based VSM (using English words). Only a few rows and columns are shown, and the values in the cells are not shown. The idea would be that in a good VSM the row vectors of *car* and *vehicle* would be much more similar to each other than to the row vector of *coffee*.

Table 1: Part of a type-based VSM.

	home	*drink*	*traffic*	*wheel*	...
car	...	...	...	...	...
vehicle	...	...	...	...	...
coffee	...	...	...	...	...
...	...	...	...	...	...

In token-based VSMs, each instance (=token) of a word in the corpus, or at least a representative number of such instances, has its own row. This is illustrated in Table 2, using, as an example, tokens of the English word *car* (please ignore, for the time being, that Table 2, contains tokens from two varieties). In token-based VSMs, it would not be a good idea to simply use the words in the vicinity of the target word as 'atomic' features (i.e. as columns). Since there are only a few context words in each token, this would lead to very sparse, very uninformative row vectors for the individual tokens. For instance, switching again to the *monitor* example from the introduction, in such an ill-chosen approach the fact that one *monitor* token has the word *kinderen* 'children' in its context and another *monitor* token has *jongeren* 'youngsters' in its context, would not lead to the desired effect of their two row vectors somehow resembling each other (since *kinderen* and *jongeren* would be treated as unrelated 'atomic' features).

What we do instead to build the row vector of a token in a token-based VSM, is for each of the context words in that token, we first retrieve its type vector, i.e. we retrieve its row vector from a type-based VSM. In order to clarify this, let us use the notation Cs for the context words that appear in a specific token. For instance, in the token *...I park my car in the second garage ...*, with *car* the target word, words such as *park*, *second*, and *garage* would be Cs. It is of these Cs that we retrieve the type vectors. Let us, in the present context, use the notation CCs for the features that are used in the type vectors of the Cs. In other words, the CCs are the (type-based) features of the (token-specific) Cs of the target word.

Having retrieved the type vectors of all Cs in a token, we add up these type vectors to build the token representation of the token (so that the CCs will become the features of the token-based VSM). For instance, the row representation of the example token just given would be the sum of the type vectors of *park*, *second*, *garage*, etc. It

should be added, though, that we use a weighted sum, in order to take into account that some Cs (e.g. *park*, *garage*) are more important than others (e.g. *second*). More specifically, the row vector of a token is the weighted sum of the type vectors of all its Cs, with the weights being the importance of the Cs, measured as the type-based 'attraction' (PPMI) between the target word and the C. A less concise description of this procedure, which is a slightly modified version of Schütze (1998), can be found in Heylen et al. (2015). Importantly, returning once again to the *monitor* example, in this approach, the row vector of a token with the C *kinderen* 'children' in it and the row vector of a token with the C *jongeren* 'youngsters' in it will tend to be similar, because the type vectors of these two important Cs can be expected to be very similar.

The above describes how token-based VSMs for a single variety can be built. Superimposed token-based VSMs for two varieties, as illustrated in Table 2, are a straightforward extension. This time we use one type-based VSM for each variety (both having the same features) and we use a sufficiently large sample of tokens from both varieties. With this, we create a matrix, as illustrated in Table 2, that has as its rows the tokens form both varieties. The row vectors are calculated as before, with this complication that that for building the row for a token from variety A, information is retrieved from the type-based VSM for A, and for building the row for a token from variety B, information is retrieved from the type-based VSM for B.

Table 2: Part of a matrix with two superimposed token-based VSMs for the target word *car*.

	CC_1	CC_2	CC_3	CC_4	...
car 1 from US	...	...	...	...	...
car 2 from US	...	...	...	...	...
car 3 from US	*weighted sum of Cs of car 3 from US*				
...	...	...	...	...	...
car 1 from UK	...	...	...	...	...
car 2 from UK	...	...	...	...	...
...	...	...	...	...	...

2.2 Token clouds in original vector space and reduced vector space

We speak of token clouds, because you can think of the information in a token-based VSM as a cloud of points (the tokens) sitting in a high-dimensional space, in which each CC is a dimension and the PPMI values are coordinates. In spite of the high-dimensional nature of such a space, it is straightforward to calculate (cosine-based) distances between the points (=tokens). However, if we want to be able to visualize the tokens cloud, we need to reduce the number of dimensions. We solve this by

applying non-metric MDS to the original matrix (in which we used cosine-based distances), in order to derive from it a 2D-simplication (in which we use Euclidean distances). We call this two-dimensional space the 'reduced vector space', as opposed to the 'original vector space' we started out with. It is this 'reduced vector space' that is represented in plots such as the ones in Figure 1.

2.3 Separation indices

In the case study in the next section, we will test four types of *separation indices*, which, as explained before, are meant to quantify the degree to which there are non-overlapping areas in the superimposed token clouds. We call the first two measures 'global' indices, because they assess to which extent the clouds as a whole tend to consist of 'larger areas' that lack overlap. The final two measures, on the other hand, are 'local' indices; they assess to which extent, at a more fine-grained level, there are smaller areas that lack overlap.

The first 'global' index, `DR`, which stands for *distance ratio*, is a slightly modified version of the clustering index proposed in McClain & Rao (1975). For each item (=token), we calculate A the mean distance from the item to other items from the same class (=variety) and B the mean distance from the item to items from the other variety. The *separation index* for the item is B/A. The *separation index* for the complete token cloud is the mean *separation index* of the items.

The second 'global' index, `SIL`, stands for *silhouette width* (Rousseeuw 1987). For each item, we calculate A the mean distance from the item to items from its own class (=variety), and B the mean distance from the item to the other class (=variety). The *separation index* for the item is $(B - A)/max(A, B)$. The *separation index* for the complete token cloud is the mean *separation index* of the items.

The first 'local' index, `SCP`, stands for (smallest) *'same class path'*. This index takes a parameter k by means of which you specific the level of granularity you want to inspect (with smaller k corresponding to more 'local' patterns). It expresses how easy it is to draw paths that connect $k+1$ same-variety tokens while encountering as few other-variety tokens as possible. The *separation index* for an item is calculated as the separation score of the shortest path of length k that connects that item to other items from the same class (=variety); the separation score of this path is the mean of the separation score of the steps it consists of; the separation score of any step from A to B is one divided by the rank of the distance of B. The *separation index* for the complete token cloud is the mean *separation index* of all items. The explanation of this index sounds complicated, and needs further explantaion, but the idea is simple. In order to determine the *separation index* for a token, the procedure tries to build a path that connects this token to k other tokens of the same variety (in one chain, stepping for A to B, from B to C, etc.) and that is as small as possible. The smaller the path that is found, the higher the *separation index* for the token. A path is small if the average length of its the individual steps is small. How small an individual step from A to B is, is determined by how many tokens from the other variety are closer to A than B is. The fewer such other-variety tokens there are, the smaller the step, and the higher its separation score. For instance, if there are no other-variety tokens

that are closer to A than B is, then the distance of B has rank 1 and therefore the separation score of the step going from A to B is $1/1$, which is 1, and which is the highest possible separation score. If there is one other-variety token closer to A than B is, then the distance of B has rank 2 and the separation score of the step from A to B is $1/2$. If there are two other-variety tokens closer to A than B is, then the rank is 3 and the separation score is $1/3$. Etc. In sum, and informally: a step is small (and therefore its separation score is high) if it doesn't cross much 'territory occupied by the other variety'. In a similar vain, the complete path of length k of a token is small (and therefore the token's *separation index* is high) if the path doesn't cross much 'territory occupied by the other variety'.

The second 'local' index is `KNN`, which stands for k *nearest neighbours*. This too is a measure that takes a parameter k by means of which one can specify a level of granularity. For each item (=token) we calculate the proportion of same-class items (=same-variety items) among its k nearest neighbours; that proportion is the *separation index* for the item. The *separation index* for the complete token cloud (i.e. all items) is the mean *separation index* for the items.

Although these four types of *separation indices* (DR, SIL, SCP, KNN) have different scales, they share a number of characteristics. Firstly, higher values indicate more presence of non-overlapping areas (so a higher degree to which the varieties occupy separate areas in the superimposed token clouds). Second, they take as their input the distances between the tokens. Since we have distances between the tokens both for the 'original vector space' and for the 'reduced vector space', we apply the *separation indices* to both.

As a result, we end up with eight sets of *separation index* calculations, since we have four types of *separation indices*, which we all apply to both the 'original vector space' and to the 'reduced vector space'. The empirical questions then are be to which extent the results will be similar across the eight sets, and, if not, how they differ.

3 Case study

We have built token clouds for 42 words, 21 of which are mentioned in several language advice resources (such as `taaltelefoon.vlaanderen.be`) as being used differently in Belgium and The Netherlands. For the other 21 words we found no such claims, nor did we have any other reason to expect semasiological regional differences. Of the former 21 words, 7 are claimed, in the language advice literature, to differ with respect to the (fixed) expressions or idioms that are often used in the two varieties, and 14 are claimed to have different possible/popular senses in the two varieties.

The following are the words that are included in the study:

- category `no` (no claims about differences found): *appel, auto, ballon, bos, broek, bureau, centrum, deur, dier, fruit, gebruiker, heling, kamer, kop, land, nacht, neus, school, steun, stoel, verlof*;

- category `expr` (expressions/idioms are claimed to differ): *biecht, boontje, geschenk, mosterd, mouw, straatje, vijg*;
- category `sense` (senses are claimed to differ): *academicus, bank, bolletje, kleedje, kous, middag, monitor, pan, patat, poep, puntje, tas, vlieger, wagen.*

For each of these words, we randomly collected 300 tokens from a large Belgian newspaper corpus (LeNC=Leuven Nieuws Corpus; 1.2 billion words) and 300 tokens from a large Netherlandic newspaper corpus (TwNC=Twente Nieuws Corpus; 500 million words), and we merged both sets in one token cloud. We used about 5000 CCs (the intersection of the top 7000 high frequency words in LeNC and TwNC, minus the top 100 high frequency words); the context window used for the type-based VSMs was 4:4 (i.e. four words to the left and four to the right of the target word). The context window for determining the Cs was 10:10. Of the candidate Cs, we only kept those that were sufficiently important according to the type-based VSM of the variety the token came from ($LLR > 1$ and $PPMI > 1$) and that also occurred in the corpus for the other variety. We dropped tokens without suitable Cs (typically retaining about 500 tokens out of the original 600). Stress in the MDS solution that we used to build the 'reduced vector spaces' varied from .15 to .28.

We then calculated the four *separation indices* (`DR`, `SIL`, `SCP`, and `KNN`), both for the 'original vector space' and for the 'reduced vector space'. In the 'local' *separation indices* `SCP` and `KNN`, k was set to 10 and an additional weighting procedure was used (that we will not go into). Finally, the resulting eight sets of *separation index* results were all standardized, in order to make it easier to compare them.

For all eight sets of *separation index* results, we ran a regression analysis with standardized *separation index* as response variable and with word category as predictor. Word category (`cat`) had the levels `no`, `expr`, and `sense`; we used dummy coding (=treatment coding), with `cat=no` as reference value. Figure 2 shows, for all eight regression models, the estimates for `cat=expr` and `cat=sense` (with 95% confidence intervals).

A few observations can be made. First, in all eight models the average *separation index* in the case of `cat=expr` is significantly higher than in the case of `cat=no`. Second, in only a few models the average *separation index* in the case of `cat=sense` is significantly higher than in the case of `cat=no`. More specifically, the latter effect is least present in models in which 'global' *separation indices* are applied to the 'original vector space', and is most clearly present in model in which 'local' *separation indices* are applied to the 'reduced vector space'.

After obtaining these results, we replicated the case study four times, with the same corpora and the same words, but each time taking other random subsets of tokens. Each time, the results were very similar; the aforementioned observations were robust across all replications.

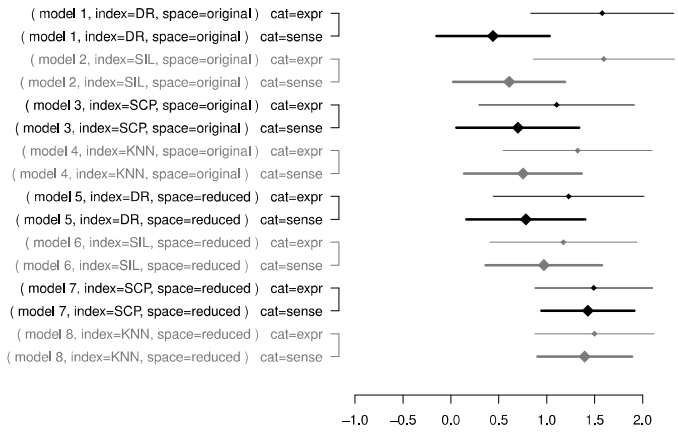

Figure 2: Estimates for `cat=expr` and `cat=sense` (with 95% confidence intervals) in 8 regression models, with reference level `cat=no`.

4 Conclusions

In this chapter, we explored the possibility of applying (dia)lectometric techniques to the investigation of (aggregate-level) semasiological variation. For such a thing to be possible, it is necessary to be able to quantify semasiological differences across language variaties. In a methodological pilot study, we tested eight different ways of quantifying semasiological differences. All approaches that were tested produced sensible results with respect to the detection of regional differences at the level of 'different (fixed) expressions or idioms'. Regional difference at the level of 'different (number of) senses', on the other hand, proved harder to detect, with the approach that applied 'local' *separation indices* to the 'reduced vector space' outperforming the other approaches. The results suggest that the dimension reduction can be instrumental in the quantitative identification of semasiological patterns in the data.

References

Cook, Paul & Graeme Hirst. 2011. Automatic identification of words with novel but infrequent senses. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, 265–274.

Gulordava, Kristina & Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of*

the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics, 67–71.

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.

McClain, John O. & Vithala R. Rao. 1975. Clustisz: a program to test for the quality of clustering of a set of objects. *Journal of Marketing Research* 12(4). 456–460.

Nerbonne, John & William Kretzschmar. 2003. Introducing computational techniques in dialectometry. *Language Resources and Evaluation* 3. 245–255.

Rousseeuw, Peter F. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics* 20. 53–65.

Ruette, Tom, Dirk Geeraerts, Yves Peirsman & Dirk Speelman. 2014. Semantic weighting mechanisms in scalable lexical sociolectometry. In Benedikt Szmrecsanyi & Bernhard Wälchli (eds.), *Aggregating dialectology, typology, and register analysis: linguistic variation in text and speech*, 205–230. Berlin/New York: De Gruyter Mouton.

Sagi, Eyal, Stefan Kaufmann & Brady Clark. 2011. Tracing semantic change with latent semantic analysis. In Kathryn Allan & Justyna A. Robinson (eds.), *Current methods in historical semantics*, 161–183. Berlin/New York: De Gruyter Mouton.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1). 97–123.

Turney, Peter D. & Patrick Pantel. 2010. From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research* 37. 141–188.