The identification of stages in diachronic data: variability-based neighbour clustering

Stefan Th. Gries[1] and Martin Hilpert[2]

Abstract

In this paper, we introduce a data-driven bottom-up clustering method for the identification of stages in diachronic corpus data that differ from each other quantitatively. Much like regular approaches to hierarchical clustering, it is based on identifying and merging the most cohesive groups of data points, but, unlike regular approaches to clustering, it allows for the merging of temporally adjacent data, thus, in effect, preserving the chronological order. We exemplify the method with two case studies, one on verbal complementation of *shall*, the other on the development of the perfect in English.

1. Introduction

Much diachronic corpus-linguistic work is based on retrieving all instances of a particular linguistic phenomenon from a corpus, coding each instance according to one or several parameters, and then charting the development of this parameter over time. As a case in point, Israel (1996) studied the development of the English *way*-construction based on data from the OED, and found that the construction increasingly requires a prepositional phrase (PP) that expresses a path. While examples like (1a) are fairly frequent in older data, modern examples of the construction typically include a PP, as seen in (1b).

(1) a. The moving legions speed their headlong way. (1715–20)
b. Mr. Bantam corkscrewed his way through the crowd. (1837)

Figure 1 shows the diachronic increase in relative frequency of instances with PPs (Israel, 1996: 225).

One central problem in the analysis of this kind of data is the question of when quantitative changes become large enough to warrant a classification into groups or stages that are also likely to be qualitatively different. From

[1] Department of Linguistics, University of California, Santa Barbara, CA 93106-3100, USA.
Correspondence to: Stefan Th. Gries, *e-mail*: stgries@linguistics.ucsb.edu
[2] International Computer Science Institute, 1947 Center Street, Berkeley, CA 94704, USA.

DOI: 10.3366/E1749503208000075
Corpora Vol. 3 (1): 59–81

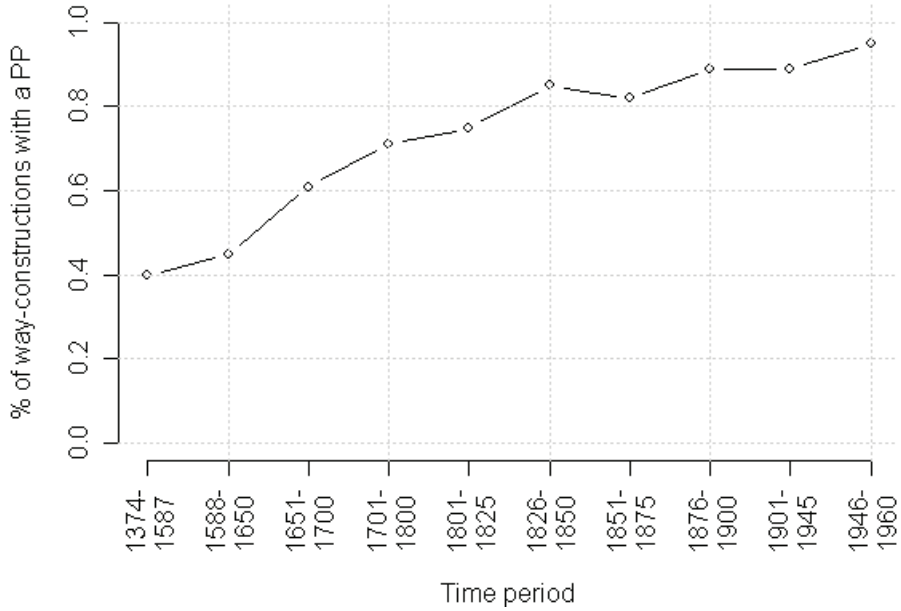

Figure 1: Increasing relative frequency of obliques in the English *way*-construction

Figure 1, it is clear that there are no immediately obvious points at which stages could be distinguished. Very often, researchers proceed on the basis of one or a mixture of the following two heuristics:

- they group their data on the basis of visual inspection of their statistics or some visual representation of the statistics; or,
- they group their data on the basis of the divisions that the data come in, established *a priori* by the corpus from which the data were extracted, that is, different corpus parts representing different temporal stages of the data.

Both of these strategies are risky. The former invites the problem of subjectivity: on the one hand, without clear operationalisations of when to assume a quantitatively different stage that may reflect something qualitatively interesting, different researchers may arrive at different groupings even for the same data set; and, on the other hand, results from different studies are likely to vary erratically such that they become difficult to compare, and overall progress becomes difficult to assess. The latter runs the risk of missing out on important generalisations: if the only division of the data that is assumed is the one that comes with the corpus data, then higher-level generalisations that only arise from grouping different temporal stages may be lost.

There is one third problem that is not often discussed, but whose omnipresence makes it worth an excursus. In both synchronic and diachronic corpus linguistics, there often seems to be an underlying (or at least unquestioned) assumption that there is one, single, reasonable way of dividing up a corpus into different parts. For example, there is a considerable body of literature on the issues of how to assess the homogeneity of a corpus and how to compare one corpus with another, which usually investigates these issues on the basis of word frequencies and corpus files (Rayson and Garside, 2000; Kilgarriff, 2001). However, the assumption of a single gold standard in structuring any corpus is false. On the contrary: the homogeneity of a corpus can only be defined with respect, specifically, to both a particular level of granularity and a particular linguistic phenomenon. This means that, to a researcher who is not interested in word frequencies but some grammatical phenomenon, the assessment of a corpus's homogeneity in terms of word frequencies is irrelevant for two reasons: first, because the homogeneity of the corpus has not been assessed on the basis of the phenomenon that the researcher is interested in but on the basis of more conveniently available word frequencies; and, secondly, because the homogeneity of the corpus with respect to the relevant phenomenon may well be extremely different – a corpus that is rather homogeneous in terms of word frequencies at the level of the register might also be less homogeneous once word frequencies are not inspected across registers, but across files or along some other dimension.

There is some empirical evidence to support this, so let us briefly illustrate this logic on the basis of two commonly-made distinctions: spoken vs. written and register distinctions. For example, Gries (forthcoming) shows that the correlations of verbs and the ditransitive construction is not meaningfully different when lemmas and inflectional verb forms are compared, and, also, the difference between speaking and writing does not result in qualitatively different conclusions. Rather, a principal components analysis of the data shows that the dimensions along which meaningful differences in the use of the ditransitive can be observed cut across the usual suspects of speaking vs. writing, but also the registers and sub-registers of the ICE-GB. Similarly, Gries (2006) shows that the phenomenon of particle placement can be predicted very well in persuasive and non-professional writing, letters and spoken public dialogue, but it can only be predicted badly in academic and non-academic writing and spoken public dialogue. These groups again cut across simplistic *a priori* groups of the corpus and show that such *a priori* divisions are not equally useful for all phenomena.

On these grounds, Gries (2006) argues that variability within and between corpora – the former having to do with homogeneity, the latter having to do with comparability – must be assessed by exploring many different divisions of a corpus at the same time, and in a phenomenon-specific way. By now, the implications for this paper are obvious: the *a priori* division of a corpus into, say, Old English, Middle English, Early Modern English and

Modern English may be meaningful for linguistic phenomenon *x*, but may not be meaningful for linguistic phenomenon *y*. Thus, whatever approach one adopts for identifying quantitatively different stages in a corpus, it must be one that can be geared specifically to the phenomenon one is interested in and must be flexible enough, if necessary, to cut across the existing corpus divisions.

One way of addressing the above problems may appear to be the use of cluster-analytic approaches and algorithms. For example, hierarchical agglomerative clustering algorithms are regularly used exploratorily in many different contexts to identify quantitatively different groups in data. The typical input to this kind of cluster analysis consists of a two-dimensional table such that the rows describe objects to be clustered into groups on the basis of the characteristics that define the columns of the table. Here is a non-linguistic example. An economist may be interested in identifying different groups of, say, 200 countries, which are listed in the rows of a two-dimensional table, on the basis of column variables such as population size, population growth, degree of urbanisation, per capita energy consumption, average household income, GNP, and so on. A different way to describe this data structure is that every object to be clustered is described by a set of ordered vector of values (for the column variables). Since the data set is too large and complex, inspecting such a table visually is not likely to yield any valuable insight, but cluster analyses allow us to perceive patterns at levels of granularity that human observers are incapable of noticing. Both within and outside linguistics, such clustering algorithms usually proceed as per the following algorithm.

```
1 compute a distance (or similarity) matrix,[3]
2 repeat
3   identify the two elements that are most similar to each other[4]
4   merge the two elements that are most similar to each other
5   compute new distances on the basis of this merger
6 until the number of elements is one
7 draw a tree structure (aka dendrogram) that summarises the groupings
  arrived at in steps 1 to 6
```

Algorithm 1: Pseudo-code of many hierarchical agglomerative clustering algorithms

While the identification of the groups is largely objective and independent of the researcher, there are two main decisions that need to be

[3] In a distance or similarity matrix, each cell of the matrix contains a number that quantifies the similarity of the element in the row to the element of the column. Depending on the particular measure that is used, distance matrices and similarity matrices can often be thought as derivative of each other. We will use the terms *distance matrix* and *distance measure* but nothing hinges on this terminological choice.

[4] In the case of ties, choose one pair randomly.

made (and defended) by the researcher.[5] On the one hand, they must decide on a *measure of variability or dissimilarity* (see line one of Algorithm 1 and the italicised phrase in line three of Algorithm 2). Usually, one distinguishes between two broad kinds of such measures, and the same kind of distinction has to be made here. First, there are measures which quantify the similarity of vectors on the basis of the distances between pairs of data points; these include Euclidean distances, Manhattan/City Block metric, and others. Thus, if the data points of two vectors are far away from each other (say, in a two-dimensional co-ordinate system), the two vectors will be considered dissimilar even if they are perfectly parallel to each other. Secondly, there are measures which quantify the similarity of vectors not on the basis of the pairwise distances, but on the overall degree of parallel curvature; these include correlation measures like Pearson's product-moment correlation, Spearman's rank correlation, and the cosine. With these measures, even if the data points of two vectors are far away from each other, if they are perfectly parallel to each other, the two vectors will be considered highly similar.

On the other hand, the researcher must decide on an *amalgamation rule*, that is, a mechanism that defines how the merging of vectors in line four of Algorithm 1 and line five of Algorithm 2 is performed. One of the most widely-used measures is Ward's method, which fuses those two vectors out of a set of vectors whose amalgamation (by averaging) results in the smallest increase in the sums of squares.

Clustering methods and related approaches have found much use in the corpus-linguistic literature (cf. Biber, 1988, 1993; Moisl and Jones, 2005; Gries, 2006, forthcoming, and many others). However, attractive as this option may at first seem, it cannot be used here for the simple reason that these algorithms are blind to the temporal order of the elements that will be grouped. As is shown in lines three and four of Algorithm 1, the algorithm just picks the two most similar data points. But, of course, that means that if for some reason a data point from Old English is quantitatively similar to a data point from Early Modern English, then the analysis may group together data as disparate as 700 years, which is of course senseless in terms of diachronic linguistics. The same logic applies to data from first language acquisition: one would not want a regular cluster analysis to group data from a child at age 1.5 years with data from the same child at age 4 – this would not make sense from the point of a developmental psychologist or psycholinguist.

Here, we propose a method of identifying groups in quantitative data that are good candidates for qualitatively – i.e., linguistically – different stages in diachronic development. The method, which is referred to as variability-based neighbour clustering (VNC), is a modification of standard hierarchical agglomerative clustering approaches and was first proposed by Gries and Stoll (forthcoming) for the analysis of data from first language

[5] For a more general and more comprehensive overview of cluster analyses, see von Eye *et al*., (2004).

acquisition. It offers the best of both worlds: on the one hand, it provides an objective, data-driven classificatory approach for temporally-ordered data that avoids the above problems of inspecting the data visually and potentially losing important generalisations. On the other hand, it does not suffer from the problem of regular clustering algorithms that fail to account for temporal ordering.

In Section 2, we will briefly explain the algorithm and discuss two short case studies in which we exemplify the algorithm by applying it to two different phenomena in the diachronic study of English. In Section 3, we will conclude, noting a few caveats and possible future applications.

2. The algorithm and two case studies

2.1 The algorithm in its general form

Variability-based neighbour clustering requires as input the kind of data represented in Figure 1, (i.e., exactly the kind of diachronic data that corpus linguists use anyway). In its most schematic form, the VNC algorithm can be represented by means of the pseudo-code in Algorithm 2.

```
Given a set of n temporally distinct corpus parts where each corpus part
(i) is named by a different (average) year and
(ii) contains one or more statistics regarding a phenomenon in question ...

1 repeat
2   for all groups of recordings named year_x and all recordings named after
    the next higher year_x+1
3     compute and store some measure of variability for the combined data of
      all recordings named year_x or year_x+1
4   identify the smallest of all n-1 measures of variability, which is called
    minvar
5   merge the data from all recordings of year_minvar and year_minvar+1
6   change the age names of all recordings of year_minvar or year_minvar+1 to the
    weighted mean of their combined years
7 until all recordings have the same year name
```

Algorithm 2: Pseudo-code of general variability-based neighbour clustering

This representation, which is perhaps slightly unusual, is to be interpreted as follows. Imagine all data points are listed next to each other in chronological order. The algorithm first accesses the first of all pairs of neighbouring data points, that is, the temporally earliest data point and the second earliest data point (line 2). It then quantifies their dissimilarity and stores that value (line 3). It does that for all pairs of data points (hence the 'for all' in line 2). Thus, if, for example, there are five data points, then lines 2–3 will yield four dissimilarity values, one for 1 and 2, one for 2 and 3, one for 3 and 4, and one for 4 and 5. Once the algorithm has done that,

it checks (line 4), which of these dissimilarity values is smallest (meaning, in turn, which similarity is largest). The data points that are most similar to one another are then merged into a new data point that contains the information of the two original data points (line 5) and gets as a name the mean of the original data points' names (line 6). If, in our example with five data points, data points 2 and 3 were most similar to each other, then a new data point would be created, which would be called 2.5 (the mean of 2 and 3) and which would contain the summarised data of the old data points 2 and 3. The new data set now contains only four data points: 1, 2.5, 4, and 5.

We have so far glossed over lines one and seven. Lines one and seven ensure that this algorithm applies recursively until the data set contains only one remaining data point. That is to say, the next step of the algorithm in our example would be to follow the stages above again, but this time on the new data set, which has only four data points. The algorithm may find that data points 4 and 5 are less dissimilar to each other than 1 is to 2.5, and 2.5 is to 4. It would then merge those to create a new data set with only three data points (1, 2.5, 4.5) and so on, until all the data have been merged and the algorithm ends, because there are no more pairs of data points to compare. This procedure will be exemplified below in more detail and with a concrete example.

It is important to note that using VNC requires the researcher to make the same two decisions as with any hierarchical agglomerative algorithm: again, one must choose a distance measure and an amalgamation rule. In each of the case studies discussed below, we will note our methodological choices. As we will point out, our main concern here is the overall algorithm, but not the exact distance measures and amalgamation rules that are used. Now, in order to render this algorithm easier to understand for those readers who are not familiar with exploratory data analysis and/or programming, and to see which kinds of results it can produce, let us see how it works on the basis of two examples.

2.2 *Shall* and its verbal complements

The first case study we investigate is concerned with the different collocational (or collostructional, see below) preferences of a syntactic pattern. Hilpert (2006) investigates the historical development of the English auxiliary verb *shall* by looking at its verbal complements. More specifically, he retrieved all instances of *shall* followed by a verb in the infinitive from the *Penn–Helsinki Parsed Corpus of Early Modern English* (PPCEME, see Kroch *et al*., 2004) and the *Corpus of Late Modern English Texts* (CLMET, see De Smet, 2005) corpora and homogenised the different spelling variants to the ModE versions. Crucially, these corpora cover six successive 70-year periods from 1500 to 1920, which Hilpert collapsed into three consecutive 140-year periods: 1500–1640, 1640–1780 and 1780–1920. For each of these

1500–1640		*1640–1780*		*1780–1920*	
Infinitive after *shall*	Tokens	Infinitive after *shall*	Tokens	Infinitive after *shall*	Tokens
be	736	*be*	557	*be*	1,074
have	291	*have*	234	*have*	527
find	133	*find*	107	*see*	239
see	131	*see*	75	*go*	195
come	120	*make*	69	*do*	176
do	117	*think*	57	*find*	116
make	94	*take*	52	*take*	95
take	92	*endeavour*	52	*make*	89
hear	73	*do*	51	*say*	87
know	69	*give*	46	*get*	82

Table 1: The ten most frequent infinitives after *shall* in three 140-year periods

three periods, Hilpert first reports the ten most frequent verbs following *shall*, which are shown in Table 1.

Since it is immediately obvious that the raw frequencies mean little here – one finds the same set of semantically general/light verbs in each period – Hilpert then computes a multiple distinctive collexeme analysis. A multiple distinctive collexeme analysis (multiple DCA) is an extension of a distinctive collexeme analysis (DCA, see Gries and Stefanowitsch, 2004). A DCA is used to compute the attraction or repulsion of words to two syntactically-defined slots in functionally-similar syntactic patterns. Consider, as an example, the well-known dative alternation in English exemplified in (2):

(2) a. He gave the book to his father.
b. He gave his father the book.

It is by now well-known that different verbs such as *give* are attracted to the constructional variants in (2) with varying strengths. For example, Gries and Stefanowitsch (2004) use a DCA to show that *give* is much more strongly attracted to the ditransitive variant in (2b) than to the prepositional dative in (2a), whereas *bring*, for instance, is much more strongly attracted to the prepositional dative than to the ditransitive. The DCA quantifies these degrees of attraction and repulsion by comparing the observed frequencies of each verb attested in the ditransitive or the prepositional dative at least once against the frequencies with which each verb would be expected to occur in the two constructions by chance. The measure of attraction or repulsion is, therefore, based on a statistical test – the Fisher–Yates exact test. However, given this statistical implementation, a DCA can only compare each word's attraction to two syntactic patterns. However, Hilpert needed to compare three historical periods and determine for each infinitive that is attested after *shall* at least once in the corpus, whether it is over- or underrepresented in

Historical period	*Obs. freq. of say*	*Obs. freq. of all verbs*	*Exp. freq. of say*	$log10\,p_{binomial}$
1500–1640	48	4,575	55.94	−0.95
1640–1780	36	3,334	40.77	−0.65
1780–1920	87	6,076	74.29	1.52

Table 2: The frequencies of *say* after *shall* in three 140-year periods

each of the three periods. He, thus, used multiple DCA as implemented in Coll.analysis 3 (see Gries, 2004). Let us explain briefly how this method works. Consider the distribution of the data for the infinitive *say* as shown in Table 2.

It is clear that *say* occurs after *shall* in the three above periods forty-eight, thirty-six and eighty-seven times. This may seem like a straightforward increase, but, of course, these raw frequencies of occurrence do not take into consideration the corpus sample sizes, and the fact that the latest corpus part, in which *say*'s frequency after *shall* is highest, is also the largest. A multiple DCA, by contrast, takes into consideration the number of tokens of infinitives after *shall* in the three periods, which are 4,575, 3,334 and 6,076 respectively by computing an exact binomial test for *say*'s frequency after *shall* in each period. To that end, one first computes the expected frequencies of *say* after *shall* in each period, which are proportional to the numbers of tokens of $shall + V_{\text{inf}}$: 55.94, 40.77, and 74.29 respectively. Then, one computes the exact probability to observe, at most, forty-eight occurrences of *shall say* in the first period when 55.94 occurrences would be expected by chance, and similarly for the two other periods and all other verbs. The output of Coll.analysis 3 is then a table which, in this case, lists for each verb the time period in which it is most strongly and least strongly attracted to the $shall + V_{\text{inf}}$ construction. For expository reasons, the p-value resulting from the binomial test is logged to the base of ten such that high values indicate strong degrees of attraction; the values are referred to as collexeme strengths. Table 3 shows Hilpert's results.

Hilpert (2006: Section 4) discusses several implications of this method of analysis, but these are not our main concern here: we are concerned with the more basic question of whether the pooling of the six 70-year stages into three 140-year stages is warranted by the data or not.

In order to investigate that consideration, we first computed a multiple DCA for all 1,201 verb lemmas that occurred at least once after *shall* in at least one of the six periods covered by the corpus data. The most relevant result from this is a table similar to Table 3: it shows all 1,201 verb types in the rows, all six time periods in the columns (not just three as in Table 3), and the logarithm to the base of ten of the p-value of an exact binomial test in each cell. These logs are positive by default and were set to negative when the observed frequency of occurrence was smaller than the expected frequency. This made it easy to distinguish between cases where a verb is attracted to a time period (the log will be positive) from cases where a

1500–1640		*1640–1780*		*1780–1920*	
Infinitive after *shall*	Collexeme strength	Infinitive after *shall*	Collexeme strength	Infinitive after *shall*	Collexeme strength
understand	15.48	*endeavor*	16.36	*forget*	17.01
come	10.32	*discover*	7.86	*go*	12.91
forfeit	6.53	*examine*	6.86	*get*	9.46
perceive	6.52	*mention*	5.90	*try*	6.87
bear	6.49	*suppose*	5.67	*meet*	6.36
appear	5.65	*confine*	5.29	*feel*	5.59
serve	5.62	*direct*	5.29	*have*	5.07
need	5.48	*explain*	5.14	*see*	4.88
eat	5.48	*think*	4.70	*write*	4.11
bring	5.28	*add*	4.33	*return*	3.96

Table 3: The ten infinitives after *shall* in three 140-year periods with the highest collexeme strengths

verb is repelled by a time period (the log will be negative). However, for our purposes, a more useful way of conceptualising the results is to say that the result of the multiple DCA was six vectors – one for each time period – of values quantifying each verb's degree of attraction or repulsion to that time period. This way of conceptualising the results already highlights the data set's similarity to the usual kind of clustering approaches: just as clustering approaches in general cluster objects on the basis of vectors describing characteristics of these objects, in this case VNC clusters six time periods on the basis of the verbs occurring (dis)preferably in them – the only difference being, again, that VNC will take the temporal order of the time periods into consideration. Consider Algorithm 3 for the pseudo-code we used.[6]

```
Given a set of m different time periods where each time period (i) is named
    by a different (average) year and (ii) contains collexeme strengths for n
    verbs ...
01 repeat
02   for all but the last historical period
03     access the vector from the x-th historical period
04     access the vector from the x+1-th historical period
05     compute and store the similarity between the two vectors (correlation
       coefficient)
06   identify the pair of time periods with the largest degree of similarity
      (i.e., the largest correlation coefficient)
07   merge the data from that pair of recordings by
08     compute the weighted pairwise means of the vector elements
09     rename the vector of the weighted pairwise means to the weighted mean
       of the original time periods
10 until all time periods have been merged
```

Algorithm 3: Pseudo-code of general variability-based neighbour clustering for the collexemes of *shall*

[6] All computations were performed with R (cf. R Development Core Team, 2007) scripts written by, and available on request from, the first author.

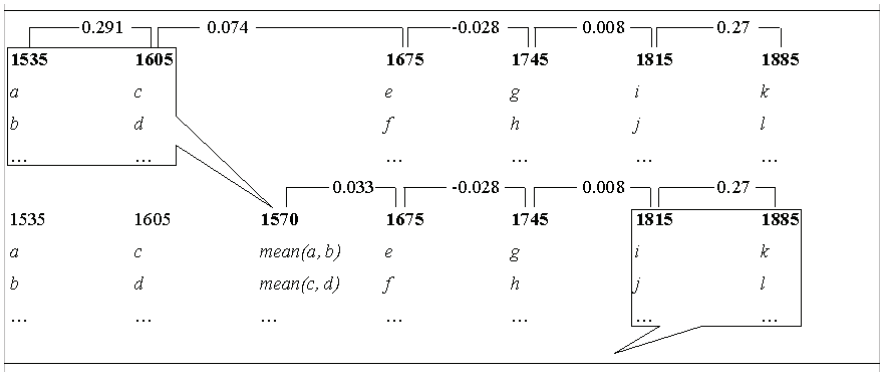

Figure 2: The first two iterations of the VNC Algorithm 3 applied to the $shall + V_{\text{inf}}$ data

Before we illustrate the procedure in more detail, two brief comments are in order. First, the similarity measure we chose is the Pearson product-moment correlation coefficient. This is because in the present case one would be more interested in similarity that is based on parallelism with regard to the curvature of the collexeme strengths rather than their absolute sizes and ranges: it more interesting to see which verbs are preferred, overall, than to see which verb scores exactly which value in which time period. Second, we amalgamate the vectors by averaging values of vectors pairwise (in a way that is analogous to the Ward method).

Let us now illustrate the application on the basis of our data and Figure 2. Algorithm 3 starts by computing the correlations between the vectors of the six time periods (lines one to five). The correlations obtained in the first iteration are shown at the top of Figure 2, with the time periods represented by their means in boldface below (that is, the time period 1500–1570 is represented by the mean/mid-point of the earliest and the latest year, i.e., 1535). The small letters in the columns defined by the historical periods are placeholders for collexeme strengths of verbs. In line six, the algorithm finds that the time periods 1500–1570 and 1570–1640 are the two most similar adjacent time periods because their correlation (0.29) is the highest of all pairwise correlations. Thus, lines seven to nine merge the vectors from these time periods by computing the pairwise means of collexeme strengths (see in the lower part the pairwise averaging of *a* and *b*, as well as *c* and *d*). The algorithm then applies again to the new data set, again computing the correlations between the vectors of the five time periods, which are also listed in Figure 2. The algorithm then finds that, this time, the last two time periods are the two most similar adjacent time periods, which are merged, and so on.

The resulting dendrogram, which plots the amalgamation of the six time periods and uses 1-correlation coefficients as the differences on the *y*-axis, is shown in Figure 3.

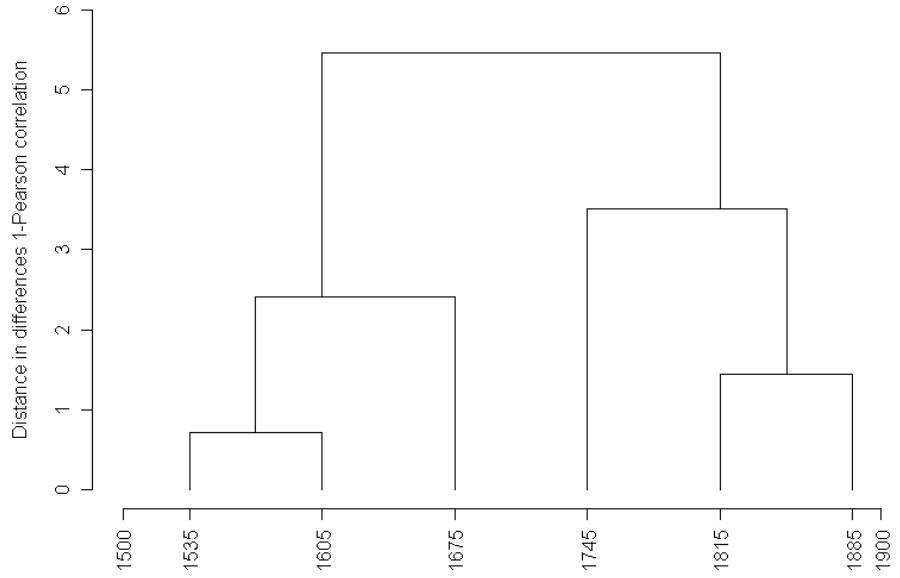

Figure 3: The VNC dendrogram resulting for the $shall + V_{\text{inf}}$ data

It is immediately clear that the data do not support a grouping of the six 70-year periods into three 140-year periods. Rather, the data fall into two 180-year groups – an early one spanning the range from 1500 to 1710 and a late one from 1710 to 1920 – but the middle two periods that were merged in Hilpert (2006) do not exhibit enough similarity, quantitatively, to make the merger appear meaningful qualitatively. This case beautifully shows how a simple decision to merge data into groups that has been made in the completely legitimate interest of increasing raw frequencies per time period can be shown to give rise to problems if the structure inherent in the data is not scrutinised beforehand. (Of course, Hilpert's overall methodological implications are still correct – it is only the theoretical/conceptual issues that may have to be revisited.)

A second aspect that is worth mentioning, at least briefly, is that the application of the VNC algorithm in this case also allows us to quantify the overall amount of structure (or noise/messiness) in the data (just as some implementations of agglomerative nesting provide an agglomerative coefficient). Let us call this measure *VC*, for variability coefficient. In this case, there were five amalgamation steps, each of which could have been associated with a maximal distance of one minus the minimal Pearson product-moment correlation r value.[7] Thus, the maximal distance the amalgamations have to deal with is $5 * 2 = 10$. In our case, the sum of all

[7] The difference 1-correlation is a standard approach to turn a Pearson product-moment correlation r, the range of which is $-1 \leq r \leq 1$, into a dissimilarity measure ranging from 0 (no dissimilarity in terms of curvature) to 2 (maximal dissimilarity in terms of curvature).

distances the algorithm had to merge – which corresponds to the highest value on the *y*-axis in Figure 3 – corresponds to 5.47. Thus, we can express the overall amount of noise as the proportion of distances that had to be merged out of the maximum possible distance, hence $VC = {}^{5.47}/_{10} = 0.547$. Obviously, this dimensionless score ranges from zero to one and, in this case, indicates a medium degree of variability in the $shall + V_{\text{inf}}$ data.

To sum up this case study, we applied one instantiation of the VNC algorithm to data on $shall + V_{\text{inf}}$ that were discussed by Hilpert (2006). The algorithm detected that while there is some overall structure in an intermediately variable data set, it is not the one proposed by Hilpert – rather, the data suggest a bipartite structure. The approach, thus, nicely illustrates how the VNC algorithm can be used to evaluate existing classifications as well as suggest new classifications for existing data. Note again in this connection that we are not suggesting that the obtained classification of the six time periods will be optimal for all sorts of linguistic phenomena. In line with our above discussion of within- and between-corpus homogeneity, our classification in Figure 3 is only meant to hold for $shall + V_{\text{inf}}$. Other lexical or other grammatical phenomena may result in a different solution, which must not be seen as a weakness of the proposed algorithm, but as a result of a finer level of resolution of one's investigation of corpora.

In the following section, we will discuss a second case study, this time looking at several formal and semantic characteristics of the development of the present perfect in English.

2.3 The development of the perfect

Our second case study is concerned with the present perfect in English. The development of this construction is of particular interest, since it involves several changes that are integral to the development of the English language at large – most importantly, including loss of morphological complexity and fixation of word order. The clustering method advanced in this paper allows us to discern stages in the development of the present perfect for exactly those changes that we choose to focus on. So, unlike in our first exemplification of the method, this case study will involve the comparison of several dendrograms. Through such comparisons, the method allows us to state whether two changes proceeded in concert, or whether they have to be viewed as distinct processes.

Searching for the tag HVP, we first retrieved all instances of the lemma *have* from the Penn corpora (YCOE, PPCME and PPCEME). Then, using tags for the participles (VBN for lexical verbs, BEN, HVN and DON for *be*, *have* and *do*), we retrieved all instances of the present perfect from these. Instances of possessive *have* and modal uses with *have to* were removed from the concordance, as were instances of the passive (e.g., *has been arrested*) or the perfect progressive (e.g., *has been singing*). The final set of matches consisted of 8,506 instances of the present perfect. In a second

EXAMPLE (abbreviated)	*PERIOD*	*GE-*	*PART-HAVE*	*ANIM.SUBJ*
for + d + am he hi h + af + d geearnad	O2	1	0	1
tet tu isehen hauest	M1	0	1	1
thou hast done grete damage unto thyselff	M4	0	0	1
but the water has overflow'd some of them	E2	0	0	0
...	...	...	...	...

Table 4: Excerpt of our data frame with the present perfect data

Time	*O2*	*O3*	*M1*	*M2*	*M3*	*M4*	*E1*	*E2*	*E2*
Freq.	186	204	441	254	857	476	1,844	2,591	1,653

Table 5: Frequencies of present perfects per time period

series of steps, we then coded each instance of the present perfect for three variables that can be used to characterise the morphological, syntactic and semantic development of the construction:[8]

- *ge*-prefixation as in *Hwat hast thou ge-don?*: 0 (no) vs. 1 (yes);
- participle before *have* as in *that he geworht hat*: 0 (no) vs. 1 (yes); and,
- subject referent animacy: 0 (no) vs. 1 (yes).

In addition, each instance came with one of the following time stamps: OE2 (for 800–1200), OE3 (1000–1150), M1 (1150–1250), M2 (1250–1350), M3 (1350–1420), M4 (1420–1500), E1 (1500–1570), E2 (1570–1640) or E3 (1640–1710). The result of this coding process was a table, a few lines of which are shown in Table 4. The frequencies of present perfects in each period are listed in Table 5.

For each variable, we expect to see a higher ratio of 'yes' responses in earlier data. In present-day English, examples with *ge*-prefixation and preposed participles are very much restricted to deliberate archaisms. An earlier study of the present perfect (Carey, 1990) showed that initial uses of the construction were restricted to animate subject referents – a restriction that we know is no longer true of present-day English. While the general asymmetry between older and more recent data is thus known, it remains

[8] Of course, these three variables do not amount to a full description of how the English present perfect changed over time. Other well-known parameters include the position of the object relative to the participle (*have the work finished* vs. *have finished the work*) and semantic traits of the verb, such as transitivity. The variables used here are selected for the purpose of illustrating the analytical method.

an empirical question just how each variable changed over the years, and whether the variables developed in comparable ways.

In order to address these questions, we applied the VNC algorithm to each vector in our data set. The specifics of the algorithm in this case study are as shown in Algorithm 4.

```
Given a set of nine different time periods where each time period (i) is
    named by a different (average) year and (ii) zeros and ones indicate the
    absence or presence of some linguistic characteristic ...

01 repeat
02   for all but the last historical period
03     access the vector from the x-th historical period
04     access the vector from the x+1-th historical period
05     compute and store the similarity between the two vectors (corrected
       means ratio cmr),[9]
06   identify the pair of time period vectors with the largest degree of
     similarity (i.e., the smallest corrected mean ratio cmr)
07   merge the data from that pair of recordings by
08     collapsing the data points of the two corresponding vectors
09     renaming the vector of the weighted pairwise means to the weighted mean
       of the original time periods
10 until all time periods have been merged
```

Algorithm 4: Pseudo-code of general variability-based neighbour clustering for the perfects

This algorithm is very similar to the previous one. The main differences are: (i) that this data set contains binary values only (zeros and ones), (ii) that the similarity measure that is used is not a correlation but the corrected mean ratio, and (iii) that the data are merged by concatenation, not pairwise averaging. Now, what results are obtained by the three clustering analyses and what do they reveal? The development of *ge*-prefixation in the present perfect is shown in Figure 4.

There are three important observations to be made. First, we observe that the dendrogram with its three main clusters mirrors the common distinction of Old English (OE), Middle English (ME) and Early Modern English (EmodE), which also underlies the categorisation of periods in the Penn Corpora. The only mismatch is the sixth period ranging from 1420 to

[9] We define a new measure here which we call the corrected means ratio, *cmr*, of two vectors v_1 and v_2. It is computed as follows:

$$cmr = (\max(\bar{v_1}, \bar{v_2}) + length(v_1 \wedge v_2)^{-1})/(\min(\bar{v_1}, \bar{v_2}) + length(v_1 \wedge v_2)^{-1})$$

This ratio divides the larger of the two means by the smaller of the two means (so that the ratio is always larger than one irrespective of the order in which the data points are compared), but only after having added to each the inverse of the combined number of data points. This addition serves to dampen the effect that ratios from small groups of data may otherwise have. Imagine the situation where the larger mean is 0.75 and the smaller mean is 0.5. The ratio of the two would then be $^{0.75}/_{0.5} = 1.5$. However, when the combined number of data points that lead to this ratio is ten, 100 or 1,000, then the dampened ratios become 1.417, 1.49 and 1.499 respectively. Thus, in effect, when the ratio was computed from a small sample, gets penalised to reduce the risk of outliers.

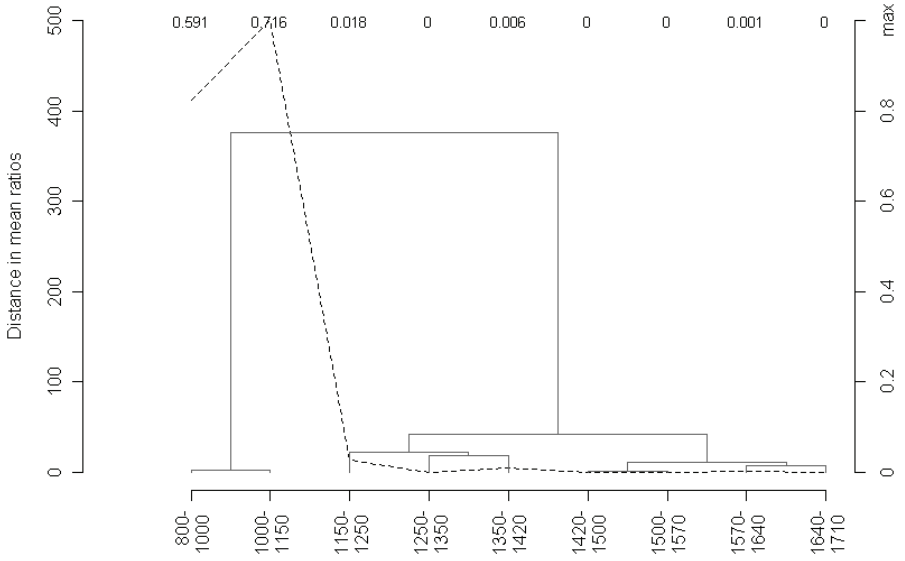

Figure 4: The VNC dendrogram resulting for the *ge*-prefixation data

1500, which belongs to the ME part of the corpus, but which is subsumed under the third and final cluster. (The good thing about this is, though, that it shows that the algorithm does not just react towards and reflect the different time periods or the different frequencies of the perfects in the different time periods.) Secondly, the dendrogram represents a binary opposition of OE and all the rest, suggesting that the most decisive development took place between OE and the beginnings of ME. Thirdly, the figures at the top are the percentages of *ge*-prefixed forms in the data for each time period, and the dotted line represents each of these percentages as proportions of the maximal percentage (which are shown on the right *y*-axis). That is, the first percentage is 0.591, which is 82.5 percent of the maximal percentage, which is 0.716, and therefore the dotted line starts at 0.825. These figures and the dotted line clearly support the interpretation following from the dendrogram: the first two time periods are obviously very different from the others. These, in turn, fall into two groups – the small proportion part that largely corresponds to ME, and the very small proportion part that largely corresponds to EModE. The figures and the dotted line also support our expectation that the number of *ge*-prefixed forms would decrease over time.

Let us now compare the dendrogram in Figure 4 to the one in Figure 5, which charts the development of the position of the participle relative to *have*. Again, three clusters approximate the distinction of OE, ME and EModE, and a substantial divide between the first cluster and the rest suggests the transition from OE to ME as the primary locus of change. Again, this is also reflected in how the proportions change over time, which, in turn,

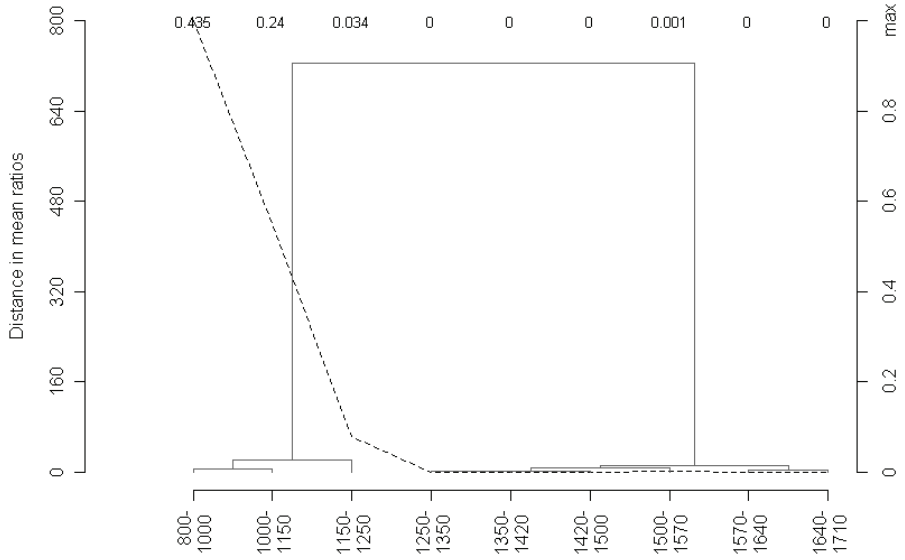

Figure 5: The VNC dendrogram resulting for the *participle before have* data

again conforms to our expectation, above, that the first three time periods are characterised by the three highest percentages whereas all other time periods exhibit extremely small percentages. Thus, while the two variables represent different grammatical domains – one being morphological, the other, syntactic – both seem to have developed in similar ways.

The similarity between the two dendrograms above stands out more clearly once they are contrasted with Figure 6, in which the semantic variable of subject animacy is represented. On the whole, while the dendrogram offers three basic clusters that are fully isomorphic to the periods of English in the Penn Corpora, it differs from the previous ones in its amalgamation of the first two clusters as well as the distribution of the feature in question. We find the decline of proportions we anticipated, but the slope/abruptness with which the decline occurs is much less steep. The EmodE period, however, stands out against the rest of the data since it contains the three smallest proportions, whereas the difference between OE and ME is noticeable to the VNC algorithm, but fairly small. All of this leads us to conclude that the development of this variable proceeded independently from the (presumably earlier) morphological and syntactic changes.

In summary, the application of the VNC algorithm to a range of variables in diachronic data allows the establishment of historical stages for the development of a construction that one chooses to study; and it allows the discrimination of distinct developments that happen across these stages. As shown above, each of the three variables produces a tree that divides into three clusters that roughly correspond to commonly-accepted periods of English. Yet, the analyses differ with respect to the grouping of the three

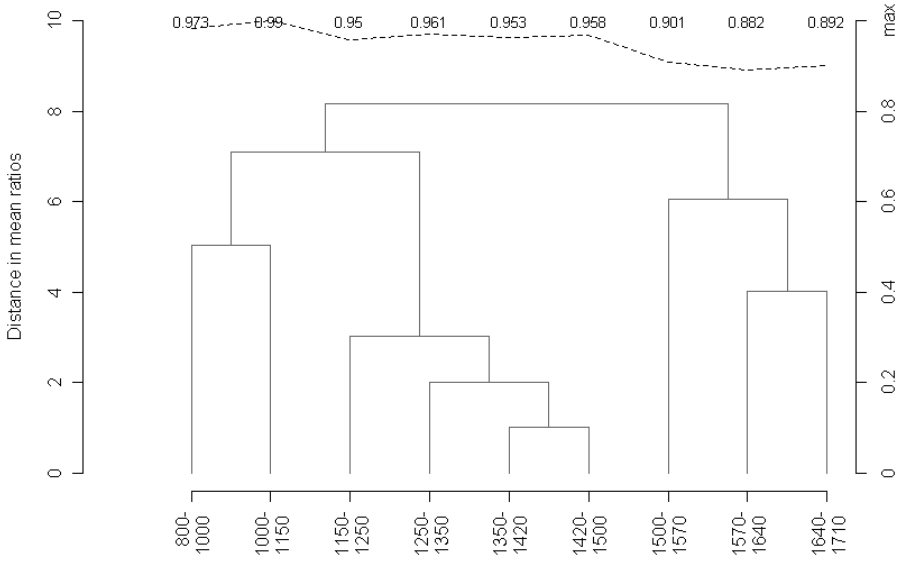

Figure 6: The VNC dendrogram resulting for the subject animacy data

clusters, with the variable of subject animacy patterning in a different way from the other two variables. In providing this kind of information, the proposed method can deepen our understanding of how grammatical change operates across grammatical domains, what grammatical structures change in concert, and how the development of a given grammatical construction unfolds over time.

3. Implications, caveats, conclusions

We hope to have shown that the kind of VNC algorithm is a useful addition to the quantitative corpus linguist's set of tools. In both case studies, VNC could be applied to data to which regular clustering algorithms could not be applied straightforwardly, and in both cases it detected structure in the data that would have been difficult for a human to detect. It is again important to point out, though, what we are saying and what we are not.

First, this paper is concerned, mainly, with a *kind* of bottom-up, data-driven approach, not with its *exact implementation*. For example, we are not saying that the Pearson product moment correlation or the proposed corrected means ratio (*cmr*) are by necessity the only or the ideal ways to quantify similarity, nor are we saying that the way we merged neighbouring vectors is by necessity the optimal way to proceed. We *are* saying, however, that a classification of the data is more useful if it is: (i) data-driven in the sense that the data suggest the groups rather than preconceptions of the researcher, and (ii) phenomenon-specific as opposed to applied conveniently across the board and on the basis of a feature other than the one that the researcher

is interested in. Note also that our approach intentionally avoids the use of significance tests. This is because we are not trying to do hypothesis-testing: as per most other clustering approaches, VNC is an exploratory approach intended to help researchers to detect structure in large/complex sets of chronologically-ordered data. Of course, if a researcher decided to follow this with formal significance testing, they can use the tree structure VNC provided for significance testing (by, for example, employing resampling methods; see Suzuki, 2006).

Secondly, in spite of what we said in the last paragraph, we are not saying that one can blindly apply VNC to one's data set and use whatever dendrogram one gets. On the contrary, we explicitly caution the reader to heed advice pertinent to the application of clustering approaches in general, as well as to the process of model building in statistics. With regard to the former, this means, in practice, that we encourage users to explore and carefully check their clusterings on the basis of different implementations of similarity measures and amalgamation rules to get a maximally clear picture of what the data look like. With regard to the latter, this means that we advise that VNC is run on the data and, if the algorithm identifies outliers, then these data points should be disregarded; VNC should then be run again to see how the clustering changes and, hopefully, improves. For users with some background knowledge in model building it may be useful to conceptualise this approach by analogy to model criticism (using, say, *AIC*). If the deletion of two out of 120 data points reduces *VC*, i.e., the overall noise in the cluster data, by 20 percent, then it is probably worth deleting these data points from the clustering and only analysing them *post hoc* to determine the cause of the exception. VNC is supposed to provide objective quantifiable suggestions for classification, but, of course, a human researcher is needed to evaluate and interpret the data. Put differently, VNC narrows down the search space for possible solutions by ruling out very many groupings that are theoretically possible, but are not supported by the data; and then it is the user who must choose and argue in favour of one of the groupings that VNC suggests. Thus, VNC is a heuristic, no more, but also no less, and we have seen, especially in Section 2.2, how revealing this kind of approach may be. Note in this connection that of course VNC can only be as precise as the data allow. If the data are only classified into six different time periods, then VNC can not, of course, provide more fine-grained results and any VNC results can only utilise this level of granularity. However, this is not a problem of the VNC algorithm – it is a general data problem: any researcher must live with pre-defined ranges if that is all that is available. Given what *is* available, though, VNC allows for building on the pre-defined ranges in ways that intuition does not.

Finally, let us briefly address a comment made by one anonymous reviewer, who suggested:

> to use an algorithm which looks for break-points in series of running averages, and then performs significance tests on the resulting segments. This would appear to be a simpler approach than described here

We disagree for three reasons. First, and as we have already mentioned, we deliberately avoid the notion of significance testing but prefer that VNC is treated as an exploratory method (just like other clustering methods). Secondly, we are not convinced that an algorithm based on break-points in a series of running averages is doubtlessly simpler. It is easy to assert that an algorithm whose exact nature is left unspecified is simpler. Furthermore, a running average approach would either require the researcher to declare a number of data points to be included in the computation of the running averages – which would render the approach less data-driven than VNC – or it would require an iterative procedure that determines the number of data points for a running average from the data. The question of whether such an iterative running-averages approach is in fact simpler than VNC, however, is an empirical question and cannot be decided by *fiat*. Thirdly, if one were not to add a second level of iterativity, this kind of approach would only provide one hierarchical level of groupings, but it would not provide the information on how, say, the second group identified by running averages would relate to the first or the third, and how the cluster of these two groups relates to the next one higher up, and so on. VNC does this automatically and is, therefore, preferable.

As with any quantitative method, there are caveats and many things to be explored. One caveat is of course that temporally-ordered corpora – be they historical or language acquisition corpora, see below – are not always particularly homogeneous. Thus, it is possible that the results are not only influenced by the linguistic phenomenon under investigation, but also by characteristics of the corpora that are used. Again, however, this is by no means a limitation of the proposed algorithm: corpus linguists always must bear in mind, and control for, confounding factors, of which matters of corpus compilation are but one. Thus, whether one applies VNC to the data or not, checking the results for such confounding variables is always part of the game, which is why careful follow-up analyses are necessary.

With regard to methodological issues, the present format of the algorithm can perhaps be improved in that it could be made more flexible. One possibility that comes to mind would involve tweaking the algorithm in such a way that it can skip the most adjacent value in case the skipped value was found to be an outlier. However, this may, superficially, appear to be an attractive option, but VNC can already handle outliers if only in a different way. When VNC is applied to a data set that has outliers, these outliers can be identified clearly: outliers will be the data points that are amalgamated rather late and at a high distance relative to the data points that surround them. This will be apparent from the vector of distances across which data points are merged and, accordingly, from long vertical arcs in the dendrogram. An algorithm that can simply skip such points ‘for now’ may merge them later at a time where their peculiarity goes unnoticed. Thus, given this perspective and the fact that we advise careful checking of potential outliers and so on anyway, we are convinced that the implementation of a skipping feature does not outweigh the close scrutiny of the data points along the above lines.

	C_1		C_2	
	cosine with one other T	cosine with the other T	cosine with one other T	cosine with the other T
T_1	T_2: *a*	T_3: *b*	T_2: *x*	T_3: *y*
T_2	T_1: *a*	T_3: *c*	T_1: *x*	T_3: *z*
T_3	T_1: *b*	T_2: *c*	T_1: *y*	T_2: *z*

Table 6: Schematic representation of a possible multifactorial VNC approach

A further area of exploration of how to refine the present form of the algorithm is concerned with extending it to handle multifactorial data differently. In both case studies, we only looked at one vector's development at a time. However, while this was part of the empirical design of the case studies – for example, in the first study, each temporal period was characterised by only one vector of collexeme strengths – data of the kind used in the second case study make it seem possible or even desirable to cluster the periods not only on the basis of one vector. More specifically, our research question in the second case study was to determine whether different variables develop in a similar way through time, so we inspected the clustering of the nine time periods independently for each variable. However, it would also be possible to try to cluster the nine different time periods on the basis of their *joint similarity* with regard to *ge*-prefixation, the position of the participle, and subject animacy. We are not yet able to put forward a fully-fledged proposal of how this could be done, but it might, for example, be possible to compare three time periods, T_1, T_2 and T_3, with respect to two characteristics, C_1 and C_2, as indicated in Table 6.

First, one compares each time period with each other time period within each characteristic (by, say, computing cosines – a similarity measure similar to correlation coefficients). That is, the similarity of T_1 and T_2 with respect to C_1 is the cosine *a*, and so on. Second, one merges those two time periods, whose cosine average is largest. That is, if time period T_1 is more similar to time period T_2 than it is to time period T_3, one merges T_1 and T_2, but not T_1 and T_3, and so on.

There are also further applications worth pursuing. First, and as mentioned above, the VNC algorithm can be applied straightforwardly to data from first language acquisition, and Gries and Stoll (forthcoming) have done just that. They used VNC to determine developmental stages on the basis of mean length of utterance values in Russian data as well as the growth of the vocabulary for an English-speaking child. Against this background, it is also possible to apply VNC to data from second language acquisition or foreign language learning to determine which, if any, clear groups the data falls into.

A somewhat more complex application might involve using VNC for the investigation of dialect continua. The overall architecture of the problem

is the same as with temporally-ordered data: a set of data points (for instance, percentages of some marker for different locations) is available and the question would be whether the distribution of the percentages supports a particular grouping of the data points. The differences from the previous applications of VNC are that the number of relevant dimensions increases from one to two (altitude and latitude), and the nature of the dimension changes from temporal to spatial. The overall logic would, however, remain the same: one would restrict a regular clustering algorithm so that only, in this case geographically, adjacent regions are clustered.

We believe that the present approach opens up a range of possibilities for research. We hope, therefore, that this paper at least motivates more researchers to at least explore more bottom-up data-driven classifications of their temporally or geographically-ordered data, and stimulates the research agenda just outlined above so that we can better come to grips with the distributional peculiarities of our trade.

References

Biber, D. 1988. Variation Across Speech and Writing. Cambridge: Cambridge University Press.

Biber, D. 1993. 'Co-occurrence patterns among collocations: a tool for corpus-based lexical knowledge acquisition', Computational Linguistics 19 (3), pp. 531–8.

Carey, K. 1990. 'The role of conversational implicature in the early grammaticalization of the English Perfect', Proceedings of the 16th Annual Meeting of the Berkeley Linguistics Society, pp. 371–81.

De Smet, H. 2005. 'A corpus of Late Modern English', ICAME Journal 29, pp. 69–82.

von Eye, A., E.Y. Mun and A. Indurkhya. 2004. 'Typifying developmental trajectories – a decision-making perspective', Psychology Science 46, pp. 65–98.

Gries, St.Th. 2004. Coll. analysis 3. A program for R for Windows 2.x.

Gries, St.Th. 2006. 'Exploring variability within and between corpora: some methodological considerations', Corpora 1 (2), pp. 109–51.

Gries, St.Th. Forthcoming. 'Corpus data in usage-based linguistics: what's the right degree of granularity for the analysis of argument structure constructions?' in M. Brda and M. Žic-Fuchs (eds) Expanding Cognitive Linguistic Horizons. Amsterdam, Philadelphia: John Benjamins.

Gries, St.Th. and A. Stefanowitsch. 2004. 'Extending collostructional analysis: a corpus-based perspectives on "alternations"', International Journal of Corpus Linguistics 9 (1), pp. 97–129.

Gries, St.Th. and S. Stoll. Forthcoming. 'Finding developmental groups in acquisition data: variability-based neighbor clustering', Journal of Quantitative Linguistics.

Hilpert, M. 2006. 'Distinctive collexeme analysis and diachrony', Corpus Linguistics and Linguistic Theory 2 (2), pp. 243–56.

Israel, M. 1996. 'The way constructions grow' in A.E. Goldberg (ed.) Conceptual Structure, Discourse and Language, pp. 217–30. Stanford: CSLI.

Kilgarriff, A. 2001. 'Comparing corpora', International Journal of Corpus Linguistics 6 (1), pp. 1–37.

Kroch, A., B. Santorini and L. Delfs. 2004. Penn–Helsinki Parsed Corpus of Early Moden English. Available online at: http://www.ling.upenn.edu/emodeng

Moisl, H. and V. Jones. 2005. 'Cluster analysis of the Newcastle Electronic Corpus of Tyneside English: a comparison of methods', Literary and Linguistic Computing 20, pp. 125–46 (supplementary issue).

R Development Core Team. 2007. 'R: a language and environment for statistical computing', R Foundation for Statistical Computing, Vienna, Austria. Available online at: http://www.R-project.org.

Rayson, P. and R. Garside. 2000 'Comparing corpora using frequency profiling', Proceedings of the Workshop on Comparing Corpora, 38th ACL, 1–6.

Suzuki, R. 2006. pvclust 1.2-0. A package for R. Accessed 8 November 2007, at: http://www.is.titech.ac.jp/%7Eshimo/prog/pvclust/