

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

Stefan Th. Gries
University of California, Santa Barbara

Martin Hilpert
ICSI Berkeley

Abstract

This paper introduces a data-driven bottom-up clustering method for the identification of quantitatively different stages in diachronic corpus data. Much like regular hierarchical clustering approaches, it is based on identifying and merging the most cohesive groups of data points, but unlike regular clustering approaches it only allows for merging temporally adjacent data, thus effectively preserving the order of events. We exemplify the method with two case studies, one on verbal complementation of *shall*, the other on the development of the perfect in English.

Key words

cluster analysis, historical development, diachrony, verbs and patterns/constructions, distinctive collexeme analysis, *shall*, perfect

1. Introduction

Much diachronic corpus-linguistic work is based on retrieving all instantiations of a particular linguistic phenomenon from a corpus, coding each instance for one or several parameters, and then charting the development of this parameter over time. As a case in point, Israel (1996) studies the development of the English *way*-construction based on data from the OED, finding that the construction increasingly requires a PP that expresses a path. While examples such as (1a) are fairly frequent in older data, modern examples of the construction typically include such 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 examples with PPs (Israel 1996:225).

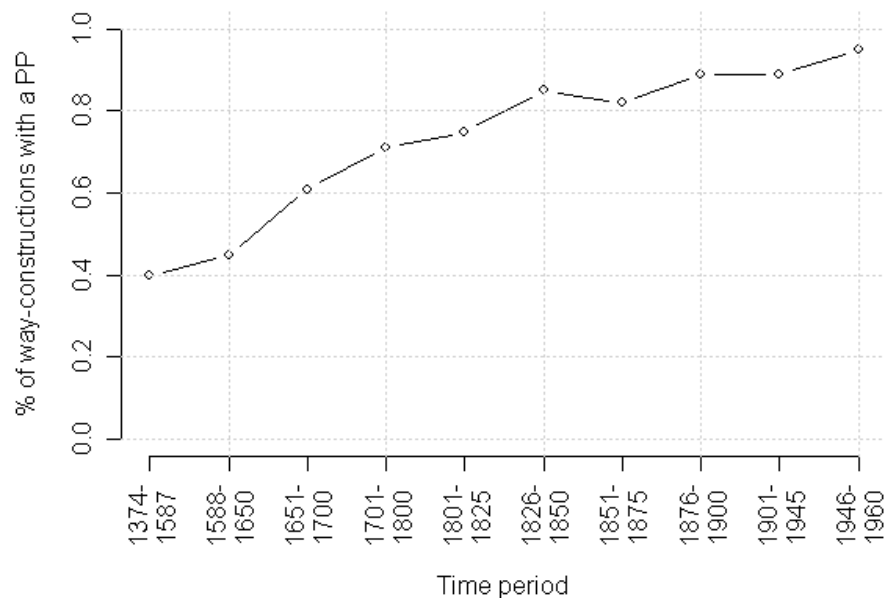


Figure 1: Increasing relative frequency of obliques in the English way-construction

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 most likely also qualitatively different. From Figure 1, it is obvious 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 eye-balling their statistics or some visual representation of the statistics;
- they group their data on the basis of the divisions that the data come in, i.e., different corpus parts representing different temporal stages of the data.

Both of these strategies are risky. The former invites the problem of subjectivity: without clear operationalizations 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, but also results from different studies are likely to vary erratically such that they become difficult to compare, and overall progress is difficult to assess. The latter runs the risk of missing out on important generalizations: if the only division of the data that is assumed is the one that comes with the corpus data, then higher-level generalizations that only arise from grouping together 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 considerably body of literature on the issues of how to assess the homogeneity of a corpus and how to compare one corpus to 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 to specifically both (i) a particular level of granularity and (ii) 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; second, 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 registers 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 (to appear) shows that the correlations of verbs and the ditransitive construction is not only not meaningfully different when lemmas and inflectional verb forms are compared, but that 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 (2007) shows that the phenomenon of particle placement can be predicted very well in persuasive and non-professional writing, letters, and spoken public dialog but it can only be predicted badly in academic and non-academic writing and spoken public dialog. 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 (2007) argued that variability within and between corpora – the former having to do with homogeneity, the latter having to do with comparability – must be assessed (i) by exploring many different divisions of a corpus at the same time and (ii) in a phenomenon-specific way. By now, the implication for the present paper is 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 to, if necessary, cut across the already existing corpus divisions.

One possibility to address 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. To provide 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 urbanization, per capita energy consumption, average household income, GNP, etc. A different way to describe this data structure is that every object that is to be clustered is described by an ordered vector of values (for the column variables). Since the data set is too large and complex, eyeballing such a table is not likely to yield any valuable insight, but cluster analysis allow us to perceive patterns at levels of granularity that human observes are incapable of noticing. Both within and outside of linguistics,

such clustering algorithms usually proceed as represented in the following algorithm.

```
1 compute a distance or a similarity matrix,1
2 repeat
3   identify the two elements that are most similar to each other2
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 summarizes the groupings
   arrived at in steps 1 to 5
```

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 made (and defended) by the researcher.³ On the one hand, one has to decide on a *measure of variability or similarity* (cf. line 1 of Algorithm 1 and the italicized phrase in line 3 of Algorithm 2). Usually, one distinguishes between two broad kinds of similarity 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 measures 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 coordinate system), the two vectors will be considered dissimilar even if they are perfectly parallel to each other. Second, there are measures which quantify the similarity of vectors on the basis of not the pairwise distances but the overall degree of parallel curvature; these measures include correlation measures such as Pearson's product-moment correlation, Spearman's rank correlation, the cosine, etc. With these measures, if the data points of two vectors are far away from each other, but perfectly parallel to each other, the two vectors will be considered highly similar to each other.

On the other hand, one has to decide on an *amalgamation rule*, i.e., a mechanism that defines how the merging of vectors in line 4 of Algorithm 1 and line 5 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 of 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 2007, to appear, and many others). However, attractive as this option may seem at first, 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 3 and 4 of Algorithm 1, the algorithm just picks the two most similar data points, but that of course means that if for whatever 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 from the point of diachronic linguistics. The same logic applies to data from first language acquisition where one would not want a regular cluster analysis to group data from a child at age 1.5 years with data from the

- 1 A distance or similarity matrix is a matrix each cell of which 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 of as derivative of each other; we will use the terms *distance matrix* and *distance measure* but nothing hinges on this terminological choice.
- 2 In the case of ties, choose one pair randomly.
- 3 For a more general and more comprehensive overview of cluster analyses, cf. von Eye, Mun, and Indurkha (2004).

same child at age 4, which would not make sense from the point of a developmental psychologist/psycholinguist.

The present paper proposes a method to identify 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 neighbor clustering (VNC), is a modification of standard hierarchical agglomerative clustering approaches and was first proposed by Gries and Stoll for the analysis of data from first language acquisition. It offers the best of both worlds: on the one hand, it offers an objective, data-driven classificatory approach for temporally-ordered data that avoids the above problems of eye-balling the data and potentially losing important generalizations. On the other hand, it does not suffer from the problem of regular clustering algorithms that it fails to account for temporal ordering.

In Section 2, we will briefly explain the algorithm and discuss two brief case studies in which we exemplify the algorithm by applying it to two different phenomena in the diachrony of English. Section 3 will summarize and point out a few caveats, conclusions, and domains of further application.

2. The algorithm and two case studies

2.1 The algorithm in its general form

Variability-based neighbor clustering requires as input the kind of data represented in Figure 1, i.e., exactly the kind of data diachronic 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 neighbor clustering

This maybe slightly unusual representation 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 neighboring data points, i.e., 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 which are most similar to each other 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, the 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 summarized 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 1 and 7. Lines 1 and 7 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 little example would be to do all the above again, but this time on the new data set with its only four data points. The algorithm may find that data points 4 and 5 are more similar 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 similarity measure and an amalgamation rule, and in each of the case studies discussed below, we will note our methodological choices. As we will point out later, our main concern here is the overall algorithm, but not the exact similarity 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; cf. 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, cf. Kroch, Santorini, and Delfs 2004) and the Corpus of Late Modern English Texts (CLMET, cf. De Smet 2005) corpora and homogenized 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 three periods, Hilpert first reports the 10 most frequent verbs following *shall*, which we report in Table 1.

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

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

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

Historical period	Obs. freq. of <i>say</i>	Obs. freq. all verbs	Exp. freq. of <i>say</i>	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

As is obvious, *say* occurs after *shall* in the three above periods 48, 36, and 87 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, 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 for *say*'s frequency after *shall* in each period an exact binomial test. 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_{inf} : 55.94, 40.77, and 74.29 respectively. Then, one computes the exact probability to observe maximally 48 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 the present case, lists for each verb in which temporal period it is most strongly and least strongly attracted to the *shall*+ V_{inf} construction. For expository reasons, the p -value resulting from the binomial test is logged to the base of 10 such that high values indicate strong degrees of attraction; the values are referred to as collexeme strengths. Table 3 lists Hilpert's results.

1500-1640		1640-1780		1780-1920	
Infinitive after <i>shall</i>	Collexeme strength	Infinitive after <i>shall</i>	Collexeme strength	Infinitive after <i>shall</i>	Collexeme strength
<i>understand</i>	15.48	<i>endeavor</i>	16.36	<i>forget</i>	17.01
<i>come</i>	10.32	<i>discover</i>	7.86	<i>go</i>	12.91
<i>forfeit</i>	6.53	<i>examine</i>	6.86	<i>get</i>	9.46
<i>perceive</i>	6.52	<i>mention</i>	5.9	<i>try</i>	6.87
<i>bear</i>	6.49	<i>suppose</i>	5.67	<i>meet</i>	6.36
<i>appear</i>	5.65	<i>confine</i>	5.29	<i>feel</i>	5.59
<i>serve</i>	5.62	<i>direct</i>	5.29	<i>have</i>	5.07
<i>need</i>	5.48	<i>explain</i>	5.14	<i>see</i>	4.88
<i>eat</i>	5.48	<i>think</i>	4.7	<i>write</i>	4.11
<i>bring</i>	5.28	<i>add</i>	4.33	<i>return</i>	3.96

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

Hilpert (2006: Section 4) discusses several implications of this way 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 question, we first computed a multiple DCA for all 1,201 verb

lemmas that occurred at least once after *shall* in at least any one of the six periods covered by the corpus data. The most relevant result from this is a table similar to Table 3: it has 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 in order to make it easy to distinguish between cases where a verb is attracted to a time period (the log will be positive) from cases where a verb is repelled by a time period (the log will be negative). However, for our purposes a more useful way of conceptualizing 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 conceptualizing 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 again being that VNC will take the temporal order of the time periods into consideration. Consider Algorithm 3 for the pseudo-code we used.⁴

```

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 neighbor clustering for the collexemes of shall

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 overall preferred 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 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 1 to 5). The correlations obtained in the first iteration are shown at the top of Figure 2, with the time periods represented by their means in bold type 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

⁴ All computations were performed with R (cf. R Development Core Team 2007) scripts written by, and available upon request from, the first author.

in the columns defined by the historical periods are placeholders for collexeme strengths of verbs. In line 6, 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 7 to 9 merge the vectors from these time periods by computing the pairwise means of collexeme strengths (cf. 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 now 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, etc. etc.

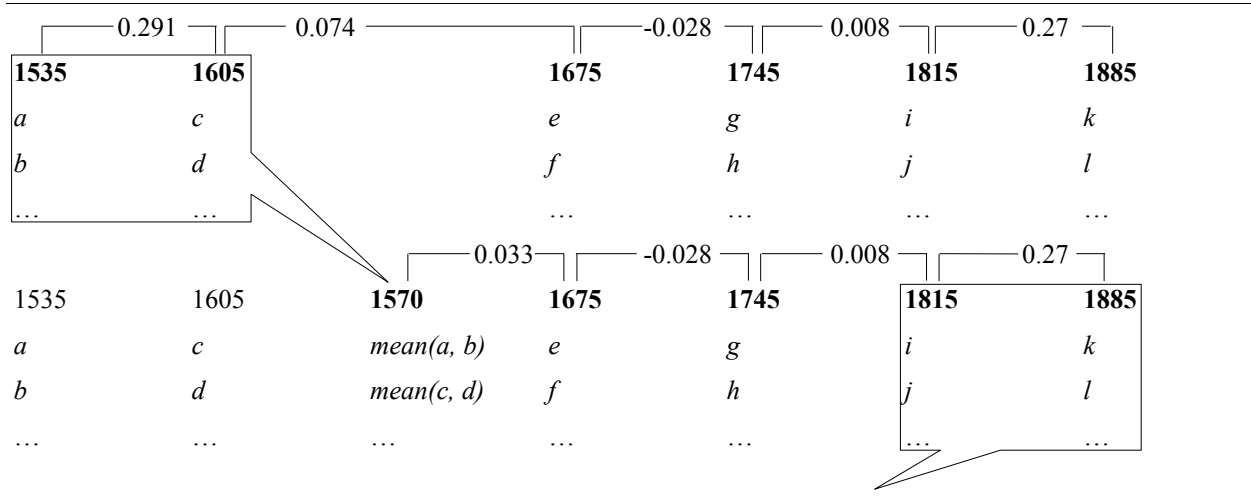


Figure 2: The first two iterations of the VNC Algorithm 3 applied to the shall+ V_{inf} data

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

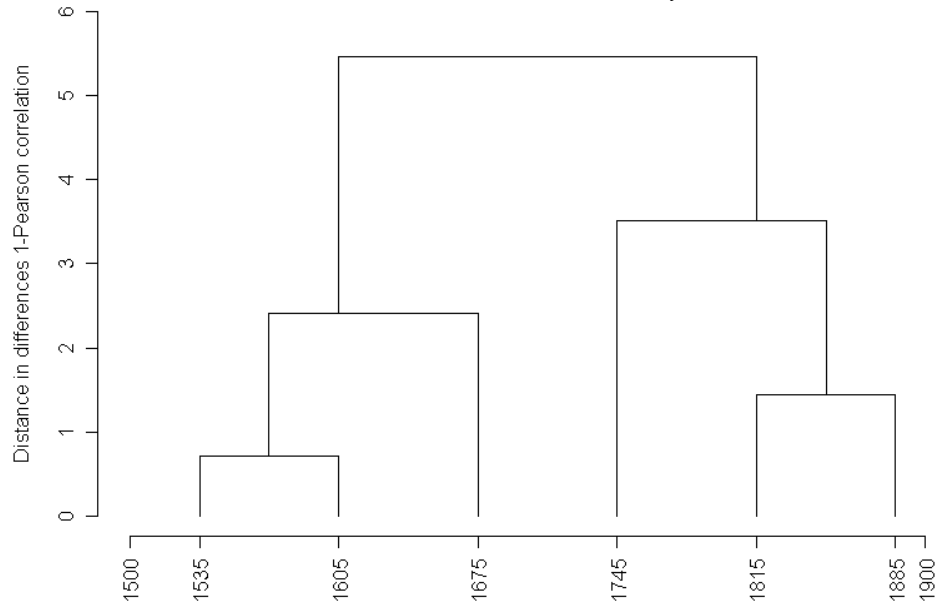


Figure 3: The VNC dendrogram resulting for the shall+ V_{inf} data

It is immediately obvious 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 scrutinized 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 1-the minimal Pearson product-moment correlation r value.⁵ Thus, the maximal distance the amalgamations have to deal with is $5*2=10$. In our case, the sum of all 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 particular case a medium degree of variability in the *shall+V_{inf}* data.

To sum up this case study, we applied one instantiation of the VNC algorithm to data on *shall+V_{inf}* that were discussed by Hilpert (2006). The algorithm detected that while there is some structure in an overall 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_{inf}* – other lexical or other grammatical phenomena may well 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*

The second case study we discuss is concerned with the present perfect in English. The development of this construction is of special interest, as 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

5 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).

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 comprised 8,506 instances of the present perfect. In a second series of steps, we then coded each instance of the present perfect for three variables that can be used to characterize the morphological, syntactic, and semantic development of the construction.⁶

- *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);
- 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.

EXAMPLE (abbreviated)	PERIOD	GE-	PART-HAVE	ANIM.SUBJ
for+d+am he hi h+af+d geeamad	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

The frequencies of present perfects in each period are listed in Table 5.

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

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) holds that

6 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.

initial uses of the construction used to be restricted to animate subject referents – a restriction that we know does no longer hold in present-day English. While the general asymmetry between older and more recent data is thus known, it remains 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 9 different time periods where each time period (i) is named
by a different (average) year and (ii) zeros and ones indicating 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)7,
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 neighbor 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.

7 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 shown in (i).

$$(i) \quad cmr = \frac{(\max(\bar{v}_1, \bar{v}_2) + \text{length}(v_1 \wedge v_2)^{-1})}{(\min(\bar{v}_1, \bar{v}_2) + \text{length}(v_1 \wedge v_2)^{-1})}$$

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 in which order 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, if the combined number of data points that lead to this ratio is 10, 100, or 1000, then the dampened ratios become 1.417, 1.49, and 1.499 respectively, effectively penalizing the ratio resulting from a small sample and reducing the risk of outliers.

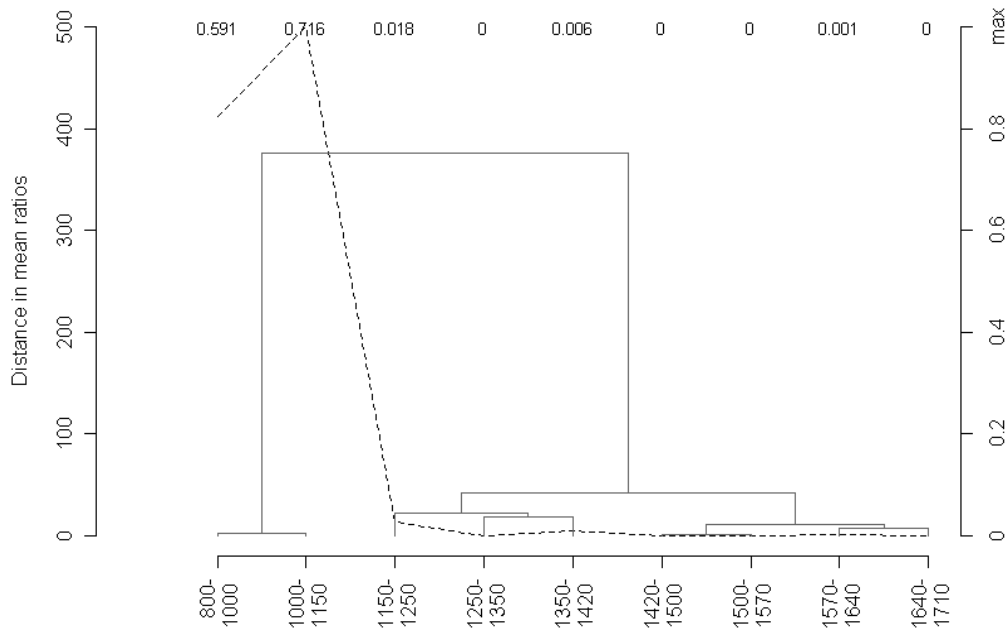


Figure 4: The VNC dendrogram resulting for the *ge*-prefixation data

There are three central 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 categorization of periods in the Penn Corpora. The only mismatch is the sixth period ranging from 1420 to 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 this shows that the algorithm does not just react towards and reflect the different frequencies of the perfects in the different time periods.) Second, the dendrogram represents a binary opposition of OE and all the rest, suggesting that the most decisive development took place between Old English OE and the beginnings of ME. Third, 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% of the maximal percentage, which is 0.716, and therefore the dotted line starts at 0.825. These figures and the dotted line not only clearly support the interpretation following from the dendrogram: the first two time periods are obviously very different from the others, which in turn fall into two groups – the small proportion part that largely corresponds to ME and the 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 again conforms to our above expectation: the first three time periods are characterized 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.

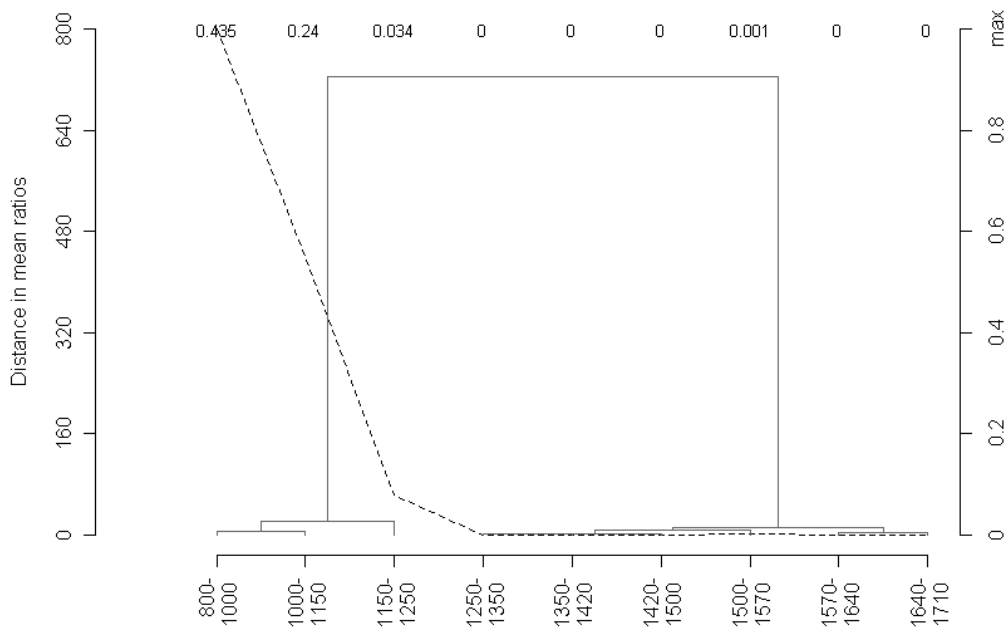


Figure 5: The VNC dendrogram resulting for the participle before have data

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 expected decline of proportions, but the slope/abruptness with which the decline occurs is much less steep. The period of EmodE, however, stands out against the rest of the data by containing the three smallest proportions, whereas the difference between OE and ME is noticeable to the VNC algorithm, but fairly small. All this warrants the conclusion that the development of this variable proceeded independently from the presumably earlier morphological and syntactic changes.

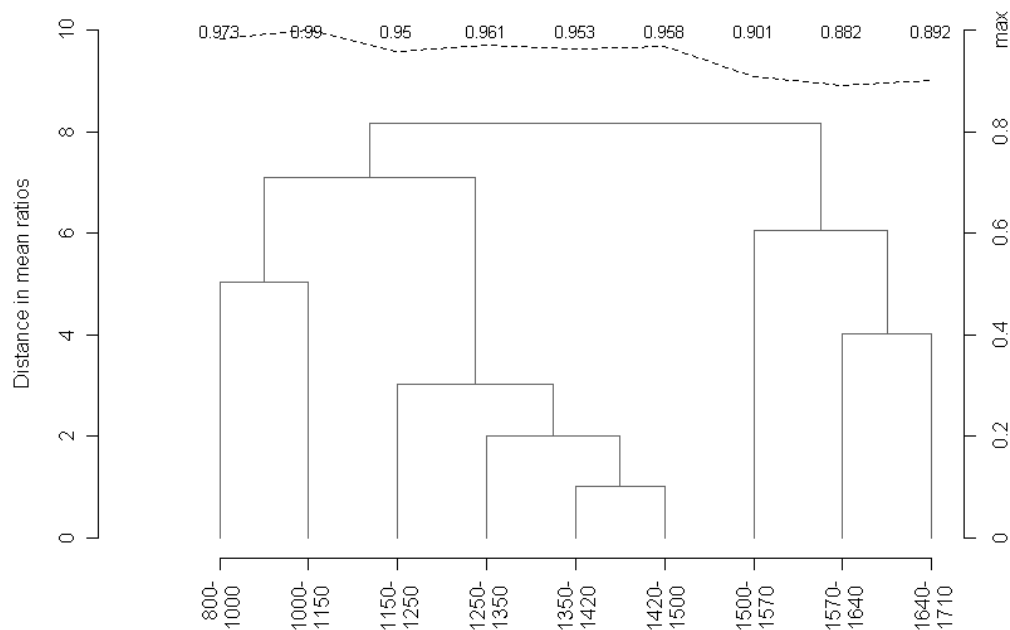


Figure 6: The VNC dendrogram resulting for the subject animacy data

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 clusters, with the variable of subject animacy patterning in a different way than the two other 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 to notice for a human observer. It is important to point out again, though, what we are saying and what we are not.

First, this paper is mainly concerned with the *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; we are not saying that the way we merged neighboring 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 one 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 – just as most other clustering approaches, VNC is an exploratory approach intended to help researchers detect structure in large/complex sets of chronologically-ordered data. Of course, if a researcher decided to follow up on this with formal significance testing, he can use the tree structure VNC provided for significance testing (by, for example, employing resampling methods; cf. Suzuki 2006).

Second, in spite of 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 (i) to the application of clustering approaches in general as well as (ii) 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 to run VNC on the data and, if the algorithm identifies outliers, then omit these data points from consideration and run VNC again to see how the clustering changes and, hopefully, improves. For users with some background knowledge in model building it may be useful to conceptualize 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%, then it is probably worth deleting these data points from the clustering and only analyze them *post hoc* to determine the cause of the exceptionality. 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, but then it is the user who has to choose and argue in favor 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 above 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 for. If the data are only classified into six different time periods, then VNC can of course not provide more fine-grained results and any VNC results can only utilize 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 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). Second, we are not convinced that an algorithm based on break-points in series of running averages is doubtlessly simpler. It is easy to assert that an algorithm whose exact nature is left unspecified is simpler. In addition, 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*. Third, 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 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, related to the next one higher up, etc. 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; cf. 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 obtained 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 the sense of being made more flexible. One possibility that comes to mind would involve tweaking the algorithm such that it can skip the most adjacent value in case the skipped value turned out to be an outlier. However, this may superficially appear like 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, then 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 for careful checking of potential outliers etc. 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.

An additional area of exploring how to refine the present form of the algorithm is concerned with extending it to handle multifactorial data differently. In both of the above 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 characterized 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 similarly across time so we inspected the clustering of the nine time periods independently for each variable. However, it would of course 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 full-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.

	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 : <i>a</i>	T ₃ : <i>b</i>	T ₂ : <i>x</i>	T ₃ : <i>y</i>
T ₂	T ₁ : <i>a</i>	T ₃ : <i>c</i>	T ₁ : <i>x</i>	T ₃ : <i>z</i>
T ₃	T ₁ : <i>b</i>	T ₂ : <i>c</i>	T ₁ : <i>y</i>	T ₂ : <i>z</i>

Table 6: Schematic representation of a possible multifactorial VNC approach

First, one compares each time period to each other time period within each characteristic (by, say, computing cosines, a similarity measure similar to correlation coefficients). That is, the similarity of T₁ and T₂ with respect to C₁ is the cosine *a*, etc. Second, one merges those two time periods, whose cosine average is largest. That is, if time period T₁ is more similar to time period T₂ than it is to time period T₃, one merges T₁ and T₂, but not T₁ and T₃, etc.

There are also further applications worth pursuing. First and as mentioned above, the VNC algorithm can straightforwardly be applied to data from first language acquisition, and Gries and Stoll (under rev.) 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 conceivable to apply VNC to data from second language acquisition / foreign language learning to determine which, if any, clear groups the data from learners fall 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 to the previous applications of VNC are that (i) the number of relevant dimensions increases from one to two (altitude and latitude) and (ii) the nature of the dimension changes from temporal to spatial. The overall logic, however, would remain the same in the sense that one would restrict a regular clustering algorithm in the sense that only, in this case geographically, adjacent regions became clustered.

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

References

- Biber, Douglas. 1988. *Variation across speech and writing*. Cambridge: Cambridge University Press.
- Biber, Douglas. 1993. Co-occurrence patterns among collocations: a tool for corpus-based lexical knowledge acquisition. *Computational Linguistics* 19.3:531-8.
- Carey, Kathleen. 1990. The role of conversational implicature in the early grammaticalization of the English Perfect. *BLS* 16, 371-81.
- De Smet, Hendrik. 2005. A corpus of Late Modern English. *ICAME-Journal* 29:69-82.

- von Eye, Alexander, Eun Young Mun, and Alka Indurkha. 2004. Typifying developmental trajectories – a decision making perspective. *Psychology Science* 46:65-98.
- Gries, Stefan Th. 2004. *Coll.analysis 3. A program for R for Windows 2.x*.
- Gries, Stefan Th. 2007. Exploring variability within and between corpora: some methodological considerations. *Corpora* 1.2:109-51.
- Gries, Stefan Th. to appear. Corpus data in usage-based linguistics: What's the right degree of granularity for the analysis of argument structure constructions? In: Brda, Mario and Milena Žic Fuchs (eds.). *Expanding cognitive linguistic horizons*. Amsterdam, Philadelphia: John Benjamins.
- Gries, Stefan Th. and Sabine Stoll. under revision. Finding developmental groups in acquisition data: variability-based neighbor clustering.
- Hilpert, Martin. 2006. Distinctive collexeme analysis and diachrony. *Corpus Linguistics and Linguistic Theory* 2.2:243-56.
- Israel, Michael. 1996. The way constructions grow. In: Goldberg, Adele E. (ed.). *Conceptual structure, discourse and language*. Stanford: CSLI, p. 217-30.
- Kilgarriff, Adam. 2001. Comparing corpora. *International Journal of Corpus Linguistics* 6.1: 1-37.
- Kroch, Anthony, Beatrice Santorini, and Lauren Delfs. 2004. Penn-Helsinki Parsed Corpus of Early Modern English. URL: <<http://www.ling.upenn.edu/emodeng/>>.
- Moisl, Hermann and Val Jones. 2005. Cluster analysis of the Newcastle Electronic Corpus of Tyneside English: a comparison of methods. *Literary and Linguistic Computing* 20 suppl.:125-46.
- R Development Core Team. 2007. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Rayson, Paul and Roger Garside. 2000 Comparing corpora using frequency profiling. *Proceedings of the Workshop on Comparing Corpora*, 38th ACL, 1-6.
- Suzuki, Ryota. 2006. pvclust 1.2-0. A package for R. URL: <<http://www.is.titech.ac.jp/%7Eshimo/prog/pvclust/>>, last accessed Nov 8, 2007.