Computational Comparison and Classification of Dialects

John Nerbonne and Wilbert Heeringa
Alfa-informatica, BCN, University of Groningen
P.O.Box 716, NL 9700 AS Groningen, The Netherlands
{nerbonne,heeringa}@let.rug.nl

July 27, 2000

Abstract

In this paper a range of methods for measuring the phonetic distance between dialectal variants are described. It concerns variants of the frequency method, the frequency per word method and Levenshtein distance, both simple (based on atomic characters) and complex (based on feature bundles). The measurements between feature bundles used Manhattan distance, Euclidean distance or (a measure using) Pearson's correlation coefficient. Variants of these using feature weighting by entropy reduction were systematically compared, as was the representation of diphthongs (as one symbol or two). The dialects were compared with each other directly and indirectly via a standard dialect. The results of comparison were classified by clustering and by training of a Kohonen map. The results were compared to well-established scholarship in dialectology, yielding a calibration of the methods. These results indicate that the frequency per word method and the Levenshtein distance outperform the frequency method, that feature representations are more sensitive, that Manhattan distance and Euclidean distance are good measures of phonetic overlap of feature bundles, that weighting is not useful, that two-phone representations of diphthongs mostly outperform one-phone representations, and that dialects should be directly compared to each other. The results of clustering give the sharper classification, but the Kohonen map is a nice supplement.

1 Introduction

In traditional dialectology, maps are divided into dialect areas on the basis of isoglosses or with the use of the arrow method (Daan and Blok 1969). However, isoglosses often simplify dialectal reality too much, not only when isoglosses fail to coincide (or even cross), but more importantly, when language varieties are dispersed through migration or war. The arrow method can only be used if dialect areas are contiguous. This paper features computational dialect comparison and classification methods. First we show three comparison methods and several variations on them, and next we show two classification methods. By using these methods, the problems signalled above are solved. On the basis of the output of the comparison methods, the dialects can be classified.

2 Dialect data

The data used for comparing dialects comes for the most part from the 'Reeks Nederlands(ch)e Dialectatlassen' (RND), which was compiled by (Blancquaert and Peé 1925–1982). From these atlases we chose 104 dialects. They were chosen to contain "easy" cases as well as difficult ones. The 104 dialects are evenly scattered over the Dutch language area (see Fig. 1), so we think they are representative of dialects in this area. In the RND for each dialect always the same 141 sentences are translated and transcribed in phonetic script. From this sentences we chose 100 words, which we think are representative for the range of sounds in the varieties.

We added the dialect of Protasovo (West Siberia, Russia). The inhabitants are Mennonites who call their dialect 'Plautdietsch'. In the 16th century, the first Mennonites moved from Southern Germany to the Netherlands and then on to Western Prussia, into the Weichsel delta area. They spoke different languages and dialects: Frisian, Low Franconian and Low Saxon. Here Siberian Plautdietsch arises as a descendant of West Prussian varieties of Low German. After 1789 the Mennonites moved to Southern Russia, and after 1870 a lot of Mennonites moved from Southern Russia to Canada and the United States (Nieuweboer and De Graaf 1994). The Siberian Plautdietsch dialect is compared to the 104 Dutch dialects to show the possibility of comparing dialects which are not contiguous but still rather similar. So in all we compare and classify 41 dialect varieties.

It would be interesting to compare Siberian Plautdietsch not only to Dutch dialects, but also with German dialects.

3 Comparison methods

In this section we identify the various comparison methods used.

3.1 Comparison of dialects

In this section three methods for comparing dialects are described. First the (phone and) feature frequency method is described, which is developed by Hoppenbrouwers and Hoppenbrouwers (1988). More details can be found in Hoppenbrouwers and Hoppenbrouwers (1993) and Hoppenbrouwers (1994). Second the Levenshtein distance is described, which was applied by Kessler (1995) to Irish Gaelic dialects with remarkable success, and by Nerbonne, Heeringa, van den Hout, van der Kooi, Otten and van de Vis (1996) who extended the application of this technique to Dutch dialects, similarly with respectable results. The Levenshtein distance is presented in (Kruskal 1983). Third the frequency per word method is described, which is a method method intermediate between the frequency method and Levenshtein distance.

Experimenting with these three methods we can see the need to regard a word as a linguistic unit (frequency vs. frequency per word) and the role of the structure of a word (frequency per word vs. Levenshtein).

3.1.1 Frequency method

With the feature frequency method frequencies of phonetic features are measured (Hoppenbrouwers and Hoppenbrouwers 1988). If this is done for two dialects, the distance between the two dialects is assessed by comparing their frequencies. A less refined method is the phone frequency method, where the frequencies of the phones are measured. The frequencies are measured by examining the 100 words. Here the 100 words form only a set of phones (about 500). For each dialect, the frequencies of the phones or of the features of the phones are divided by the total number of phones, so we get relative frequencies.

3.1.2 Levenshtein method

The Levenshtein distance may be understood as the cost of (the least costly set of) operations mapping one string to another. The basic costs are those of (single-phone) insertions, deletions and substitutions. Insertions and deletions costs half that of substitutions. The simplest versions of this method are based on calculation of phonetic distance in which phonetic overlap is

binary: nonidentical phones contribute to phonetic distance, identical ones do not. Thus the pair [a,p] counts as different to the same degree as [b,p]. In more sensitive versions phones are compared on the basis of their feature values, so the pair [a,p] counts as much more different than [b,p]. (NERBONNE ET AL. 1996) and (NERBONNE AND HEERINGA 1997) contain details explaining the application of Levenshtein distance to dialect data.

For calculating the distance between two dialects 100 Levenshtein distances are determined—one difference per word. If we simply use the Levenshtein distance, it tends to bias measurements so that changes in longer words tend to contribute more towards the average phonetic distance (since they tend to involve more changes). This may be legitimate, but since a word is a crucial linguistic unit we chose to stick to average word distance. This involves the computation of 'relative distance', which we get by dividing the absolute distance by the average length of the two words.

There are two important differences between the frequency method and the Levenshtein method. First the frequency method does not attach any significance to words, while the Levenshtein method applies one word at a time, and second, the frequency method is not sensitive to structural differences such as the order of sounds, while the Levenshtein method considers for each word its structure.

3.1.3 Frequency per word method

With the frequency per word method the frequencies of phones or of features of phones are determined per word. Two words are compared by comparing the frequencies of phones or features within them. For each word the frequencies of phones or features of phones are divided by the total number of phones, so we get relative frequencies.

The difference between the frequency method and the frequency per word method is that the frequency method does not attach any significance to words, while the frequency per word method considers words as separate entities. The difference between the frequency per word method and the Levenshtein method is that the frequency per word method is not sensitive to the order of sounds in a word, while the Levenshtein method considers for each word its sequential structure.

3.2 Symbols or features

If we compare dialects on the basis of phonetic symbols, it is not possible to take into account the affinity between sounds that are not equal, but are still

kindred. As mentioned earlier, methods based on phonetic symbols will not regard the pair [b,p] as more kindred than [a,p]. This problem can be solved by replacing each phonetic symbol by a bundle of features. Each feature can be regarded as a phonetic property which can be used for classifying of sounds. A feature bundle contains for each feature a value which indicates to what extent the property is instantiated. Since diacritics influence feature values, they likewise figure in the mapping from transcriptions to feature bundles, and thus automatically figure in calculations of phonetic distance.

In our experiments, we have used two feature systems. The one is described in Hoppenbrouwers' publications (HOPPENBROUWERS AND HOPPENBROUWERS 1988), (HOPPENBROUWERS AND HOPPENBROUWERS 1993) and (HOPPENBROUWERS 1994). The other can be found in (VIEREGGE, RIETVELD AND JANSEN 1984). Hoppenbrouwers' system is based on Chomsky and Halle's Sound Pattern of English and consists of 21 binary features which all apply for all phones (vowels and consonants). Vieregge's system consist of 4 multi-valued features only for vowels, and 10 multi-valued features only for consonants. Vieregge's system was developed for a similar comparison task, that of checking the quality of phonetic transcription. This involves comparison to consensus transcriptions.

3.3 Representation of diphthongs

Dutch has a rich system of diphthongs, which, moreover have been argued to be phonologically disegmental (MOULTON 1962) or, alternatively monosegmental (HOPPENBROUWERS AND HOPPENBROUWERS 1988). We therefore experimented with single-phone and two-phone representations. Sometimes the one, sometimes the other seems to be preferable.

3.4 Comparison of feature bundles

First we describe how feature bundles or histograms of feature values can be compared. Next we will describe some special cases.

3.4.1 Methods

We compare three methods for measuring phonetic distance between feature bundles or histograms of feature values.

The first is Manhattan Distance, also called 'taxicab distance' or 'city block distance' (JAIN AND DUBES 1988). This is simply the sum of all feature value differences for each of the n features.

$$\delta(X,Y) = \sum_{i=1}^{n} |X_i - Y_i|$$

Second we tried Euclidean Distance (JAIN AND DUBES 1988). As usual, this is the square root of the sum of squared differences in feature values.

$$\delta(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$$

Third we examined the Pearson correlation coefficient, r (BUYS 1989). To interpret this as distance we used $1 - r$, where

$$r(X,Y) = \frac{1}{n} \sum_{i=1}^{n} z_{x_i} z_{y_i}$$

and where z_{x_i} and z_{y_i} are the normalized values of the vector (at the i-th position.

3.4.2 Special cases

By using the feature system of Hoppenbrouwers all phones can be compared with each other. By using the feature system of Vieregge only vowels can be compared with vowels and consonants with consonants. For the frequency method and the frequency per word method this means that we use two separate histograms, one for vowels and one for consonants. The distance between two dialects respectively words is the sum of the difference between two histograms for vowels and the difference between two histograms for consonants.

If we use the frequency per word method and two words are compared, it is possible that one word contains one or more vowels and the other none, or that one word contains one or more consonants and the other none. In the case of vowels, the difference between the vowel histograms is equal to the greatest possible difference between two vowels, and in the case of consonants, the difference between the consonant histograms is equal to the greatest possible difference between two consonants.

If we apply the Levenshtein method we choose a fixed value as distance between a vowel and a consonant, namely the sum of the greatest possible difference between two vowels and the greatest possible difference between two consonants.

In the Levenshtein-algorithm based on symbols, three operations were used: 'substitution', 'insertion' and 'deletion'. A substitution was regarded as a combination of an insertion and a deletion, so 'substitutions' counted two and 'indels' one. When we compare feature bundles instead of phonetic symbols, the value for a substitution is no longer a fixed value, but varies between two extremes. However, for indels we have to choose a fixed value as well. Therefore we choose the greatest possible difference between two phones and divide it by two.

3.5 Weighting of features

Not all features of sounds are equally important in comparing dialects. For example it turned out that no positive value for the feature [flap±] occurred in any of the sounds of the words in the dialects examined. We therefore experimented with weighting each feature by information gain, a number expressing the average entropy reduction realized when a feature's value is known (QUINLAN 1993).

Assume k different phonetic segments appear in the dialect data: $S_1..S_k$. All dialects together contains $|D|$ phones, so we have $|D|$ feature bundles. $|Sj|$ feature bundles belong to segment S_j. Now we can calculate the database information entropy:

$$H(D) = -\sum_{j=1}^{k} \frac{|S_j|}{|D|} \times log_2(\frac{|S_j|}{|D|})$$

This may be interpreted as the average number of bits of information required to determine which segment is being used.

The number of feature bundles where feature f has value v_i is $|D_{[f=v_i]}|$. Now we can calculate the average entropy per feature:

$$H(D_{[f]}) = \sum_{v_i \epsilon V} \frac{|D_{[f=v_i]}|}{|D|} \times H(D_{[f=v_i]})$$

The average is weighted by the frequency of the feature taking on a particular value. For calculating $H(D_{[f=v_i]})$ the first formula is used, applied to the set of feature bundles which have value v_i for feature f. The information gain associated with feature f is then:

$$G(f) = H(D) - H(D_{[f]})$$

We examine versions of distance weighting where features are weighted with (multiplied by) information gain.

We note that we investigated, but could not apply a more sensitive measure, gain ratio. Gain ratio is obtained by normalizing information gain by dividing it by the number of bits required to determine wich value the feature is taking on. The latter is just the entropy of the distribution of values of a single feature in the database:

$$split(f) = -\sum_{v_i \epsilon V} \frac{|D_{[f=v_i]}|}{|D|} log_2 \sum_{v_i \epsilon V} \frac{|D_{[f=v_i]}|}{|D|}$$

Gain ratio divides $G(f)$ by $split(f)$. But gain ratio is used in machine learning where features don't determine classifications perfectly. Since phonetic features do classify features perfectly, gain ratio doesn't distinguish them.

3.6 Direct or indirect comparison

If we have three dialects A, B and C, we can calculate the distances between A and B, A and C, and B and C. In that case we get a symmetric 3×3 matrix of distances. If we have a standard dialect S and three dialects A, B and C, we can calculate the differences between the frequencies of phones or features, or the distances between words of S and A, S and B, and S and C. In that case, we get for each dialect a vector of frequency differences or word distances.

Note that from a 3×3 matrix a 3-dimensional vector can be constructed for each dialect, where each element is a distance to one of the other two dialects (one of which will be zero—representing the distance of the dialect to itself). From feature bundles of frequency differences or word distances a 3×3 matrix can be constructed, by calculating the distances between the feature bundles.

In our experiments, we analyzed 41 dialects instead of three. As standard dialect we chose standard Dutch.

4 Classification methods

The results of the different comparison methods consist of a 41×41 matrix, or of 41 vectors, where each vector corresponds to a dialect (with 41 dimensions corresponding to distances to dialects). On the basis of the matrix the dialects can be clustered. The final result is a dendrogram, a tree where the dialects are the leafs (see Fig. 2). With the feature bundles a Kohonen map can be trained (see Fig. 3). The final result is a map, where the geographic

distance between kindred dialects is small, and between different dialects great.

4.1 Clustering

Clustering is commonly applied in history ((Boonstra, Doorn and Hendrickx 1990), pp. 143 ff.), but also finds application in psycholinguistics ((Woods, Fletcher and Hughes 1986), pp. 249 ff.). It is most easily understood procedurally. At each step of the procedure we select the shortest distance in the matrix, and then fuse the two data points which gave rise to it. Since we wish to iterate the procedure, we have to assign a distance from the newly formed cluster to all remaining points. This is done by a matrix-updating algorithm. Jain and Dubes mention seven matrix updating algorithms (Jain and Dubes 1988). In our experiments, the Ward's method (or minimum variance) turned out to be most suitable. If the distance between i and j is minimal, then we form a cluster $[i, j]$ and calculate the distance from each k to the new cluster:

$$\begin{aligned} d_{k[ij]} &= ((n_k + n_i)/(n_k + n_i + n_j)) * d_{ki} \\ &+((n_k + n_j)/(n_k + n_i + n_j)) * d_{kj} \\ &-(n_k/(n_k + n_i + n_j)) * d_{ij} \end{aligned}$$

where n_i, n_j and n_k are the number of dialects belonging to respectively cluster i, j and k, and d_{ij}, d_{ik} and d_{jk} are the distances between respectively i and j, i and k, and j and k.

4.2 Kohonen maps

One of the properties of the human brain is that it is able to associate concepts with categories. On a computer, this can be simulated by implementing a Kohonen map (Kohonen 1988). When a Kohonen map is trained with concepts, each concept is assigned to a location on the map, and like concepts are located near each other while unlike concepts are located farther away from each other.

Each dialect is represented as a vector of frequency differences, word distances or dialect distances (input vector). The Kohonen map consists of a layer of cells. Each cell has a weight vector with for each feature of the input vector a weight. If the Kohonen map is initialized, a random value is assigned to each weight of the vector of each cell. The training of the Kohonen map consists of a number of epochs. In each epoch all input

vectors are read and processed in arbitrary order. The processing consists of two steps, namely selecting the winner and updating the surroundings of the winner ((KLEIWEG 1995), (KLEIWEG 1996)).

Selecting the winner means that the cell whose weight vector is most like the input vector is chosen as the winner. Updating the surroundings of the winner implies that the weight vector of the winner and the weight vectors of the cells around the winner are modified to be more like the input vector. During the epochs the size of the surroundings of winner is continuously reduced, and the extent to which the weight vectors are modified is also continuously reduced.

A Kohonen map can be combined with a minimal spanning tree. Constructing a minimal spanning tree is like clustering. At each step of the procedure we select these two vectors (the one from one, and the other from another cluster) whose distance is shortest. Next a line is drawn between the two vectors. If more pairs of vectors (each from one of the two clusters) have the same distance, the vectors of that pairs are also connected. Finally the clusters are combined to one cluster (KLEIWEG 1996).

4.3 Results

For evaluating the results we can use the map of Daan (DAAN AND BLOK 1969) which is constructed with the arrow method. People were asked to name the places near their town where (almost) the same dialect is spoken, and those where an absolutely different dialect is spoken. Next, places with (almost) the same dialects were connected with arrows and so dialect areas emerged. The map divides the Dutch language area into 28 areas of roughly comparable dialect regions, so we can see which dialects should belong to the same area. The classifications obtained by compuational methods were compared to this map to assess their probity. We reach the following conclusions:

- The results of the frequency per word method are better then the results of the frequency method. So the frequency per word method shows that a word should be regarded as a linguistic unit. The Levenshtein method also outperforms the frequency method, but is comparable to the frequency per word method, so we cannot conclude that the internal structure of words needs to be taken into account.

- Phones can be represented as symbols, or as feature bundles according to Hoppenbrouwers' or Vieregge's system. In all cases good results can

be obtained, but comparing methods shows Hoppenbrouwers' features to be more sensitive.

- Two-phone representations of diphthongs mostly outperform single-phone representations.
- For comparing histograms and feature bundles Manhattan distance and Euclidean distance are useful. Both outperform Pearson distance.
- Weighting of features mostly improve the results.
- Comparing dialects directly with each other outperforms comparing dialects via a standard dialect.
- Dendrograms show a very sharp classification (see Fig. 2). A classification visualized by a Kohonen map is vaguer (see Fig. 3). Kohonen maps cannot replace dendrograms but supplement them.

5 Acknowledgements

We thank Peter Kleiweg for his graphic programs (seen in all the figures here). Also we thank Rogier Nieuweboer for providing phonetic texts in Plautdietsch and for extensive explanations about it.

Figure 1: The locations of the 104 Dutch dialects studied.

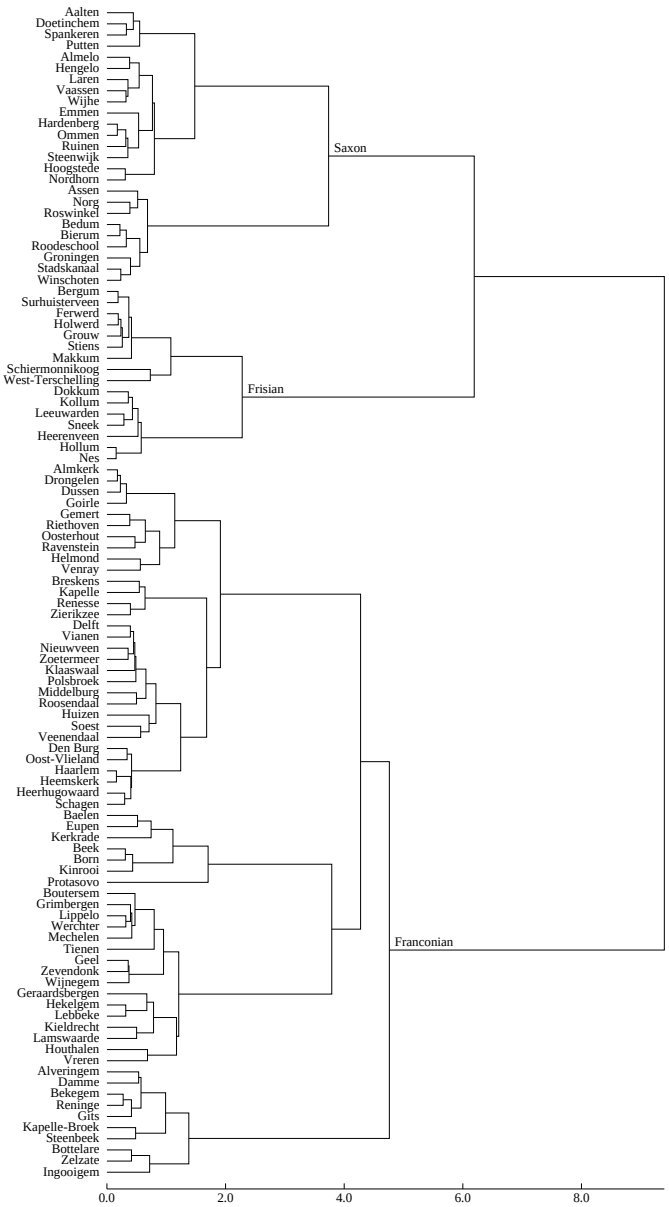

Figure 2: A dendrogram derived from the distance matrix based on Levenshtein distances. The feature system of Hoppenbrouwers is used; diphthongs are represented as two phones; Euclidean distance between feature bundles is calculated; and the dialects are directly compared with each other. The three main groups are the Franconian (lowest), Frisian (middle) and Saxon dialects (top). This accords well with dialectal scholarship. Note that Protasovo (Siberian Plautdietsch) should belong to the Low Franconian group. In particular it is grouped with the dialects that are closest to German.

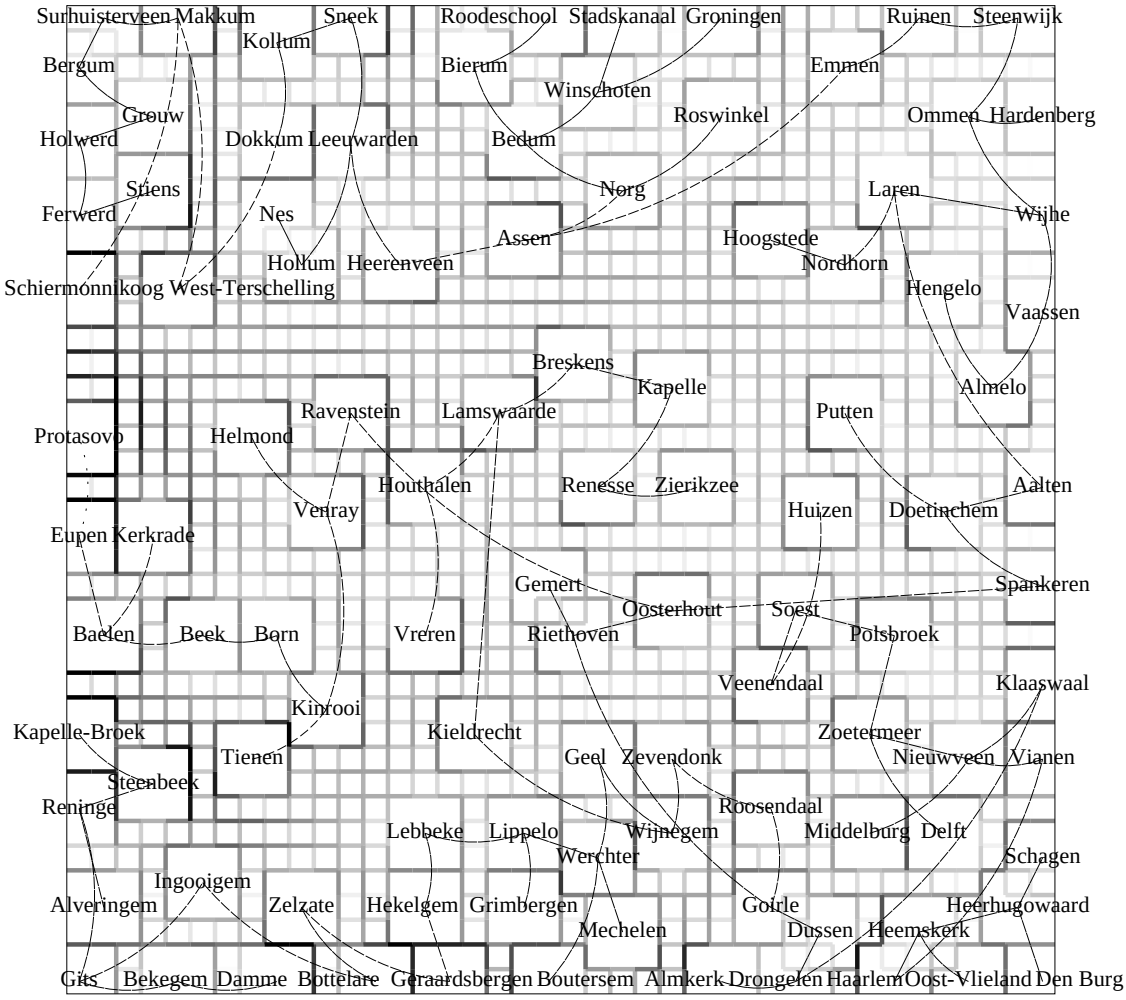

Figure 3: A Kohonen map is trained with feature bundles based on Levenshtein distances. The feature system of Hoppenbrouwers is used; diphthongs are represented as two phones; distance between feature bundles is calculated using Euclidean distance; features are not weighted; and the dialects are directly compared with each other (and not to a standard). The lines in the grid indicate the distance between dialects. The darker the line the greater the distance between the cells on the two sides. The map is overlaid with a minimal spanning tree, where more solid branches show close distances. The upper left corner is Frisian, the upper right Saxon, and, in the remaining Franconian sections, Flemish en Limburg dialects are left of Dutch dialects.

References

Blancquaert, E. / W. Peé (1925–1982): Reeks Nederlands(ch)e Dialectatlassen. Antwerpen: De Sikkel.

Boonstra, O., P. Doorn / F. Hendrickx (1990): Voortgezette statistiek voor historici. Muiderberg: Coutinho.

Buys, A. (1989): Statistiek om mee te werken. Leiden en Antwerpen: Stenfert Kroese.

Daan, J. / D. P. Blok (1969): Van Randstad tot Landrand. Toelichting bij de kaart: Dialecten en Naamkunde. Amsterdam: Noord-Hollandsche Uitgevers Maatschappij.

Hoppenbrouwers, C (1994): De indeling van de zuidoostelijke steektalen. In: TABU: Bulletin voor taalwetenschap **24**(2), 37–63).

Hoppenbrouwers, C. / G. Hoppenbrouwers (1988): De feature-frequentiemethode en de classificatie van nederlandse dialecten. In: TABU: Bulletin voor taalwetenschap **18**(2), 51–92.

Hoppenbrouwers, C. / G. Hoppenbrouwers (1993): De indeling van noordoostelijke dialecten. In: TABU: Bulletin voor taalwetenschap **23**(4), 193–217.

Jain, A. K. / R. C. Dubes (1988): Algorithms for clustering data. Englewood Cliffs, New Yersey: Prentice Hall.

Kessler, B. (1995): Computational Dialectology in Irish Gaelic. Dublin: EACL. In: Proceedings of the European Association for Computational Linguistics. 60–67.

Kleiweg, P. (1995): Practicum Topgrafische Kaarten. Groningen: University of Groningen. Available as: http://odur.let.rug.nl/~nerbonne/teach/neuro/kleiweg/kohonen.html.

Kleiweg, P. (1996): Neurale netwerken; een inleidende cursus met practicum voor de studie alfa-informatica. Master's thesis. Groningen: University of Groningen. Available as: http://odur.let.rug. nl/~nerbonne/teach/neuro/kleiweg/nn.html#practica.

Kohonen, T. (1988): Self-organization and associative memory. Berlin: Springer Verlag.

Kruskal, J. B. (1983): An overview of sequence comparison. In: D. Sankoff / J. Kruskal (eds.): Time Warps, String edits, and Macro molecules. The sequence theory and practice of sequence comparison. Massachusetts: Addison-Wesley, pp. 1–40.

Moulton, W. (1962): The vowels of dutch: Phonetic and distributional classes. In: Lingua **11**, 294–312.

Nerbonne, J. / W. Heeringa (1997): Measuring Dialect Distance Phonetically. In: J. Coleman (ed.): Workshop on Computational Phonology. Madrid. Available as: http://odur.let. rug.nl/ ~nerbonne/paper.html.

Nerbonne, J. / W. Heeringa / E. van den Hout / P. van der Kooi / S. Otten / W. van de Vis (1996): Phonetic Distance between Dutch Dialects. In: G. Durieux, W. Daelemans & S. Gillis (eds.): CLIN VI, Papers from the sixth CLIN meeting. Antwerp: University of Antwerp, Center for Dutch Language and Speech, pp. 185–202. Available as: http://odur.let.rug.nl/ ~nerbonne/ paper.html.

Nieuweboer, R. / T. De Graaf (1994): The language of the West Siberian Mennonites. In: RASK; Internationalt tidsskrift for sprog og kommunikation pp. 47–61.

Quinlan, J. R. (1993): C4.5: Programs for Machine Learning. San Mateo, California: Morgan Kaufmann Publishers.

Vieregge, W. H. / A. C. M. Rietveld / C. I. E. Jansen (1984): A Distinctive Feature Based System for the Evaluation of Segmental Transcription in Dutch. In: Proceedings of the 10th International Congress of Phonetic Sciences. Dordrecht, pp. 654–659.

Woods, A. / P. Fletcher / A. Hughes (1986): Statistics in Language Studies. Cambridge: Cambridge University Press.