Recognizing Groups Among Dialects

Jelena Prokić John Nerbonne

Abstract

In this paper we apply various clustering algorithms to the dialect pronunciation data. At the same time we propose several evaluation techniques that should be used in order to deal with the instability of the clustering techniques. The results have shown that three hierarchical clustering algorithms are not suitable for the data we are working with. The rest of the tested algorithms have successfully detected two-way split of the data into the Eastern and Western dialects. At the aggregate level that we used in this research, no further division of sites can be asserted with high confidence.

1 Introduction

Dialectometry is a multidisciplinary field that uses various quantitative methods in the analysis of dialect data. Very often those techniques include classification algorithms such as hierarchical clustering algorithms used to detect groups within certain dialect area. Although known for their instability (Jain and Dubes, 1988), clustering algorithms are often applied without evaluation (Goebl, 2007; Nerbonne and Siedle, 2005) or with only partial evaluation (Moisl and Jones, 2005). Very small differences in the input data can produce substantially different grouping of dialects (Nerbonne et al., 2008). Without proper evaluation, it is very hard to determine if the results of the applied clustering technique are an artifact of the algorithm or the detection of real groups in the data.

The aim of this paper is to evaluate algorithms used to detect groups among language dialect varieties measured at the aggregate level. The data used in this research is dialect pronunciation data that consists of various pronunciations of 156 words collected all over Bulgaria. The distances between words are calculated using Levenshtein algorithm, which also resulted in the calculation of the distances between each

two sites in the data set. We apply seven hierarchical clustering algorithms, as well as the k-means and neighbor-joining algorithm to the calculated distances and examine these using various evaluation methods. We evaluate using several external and internal methods, since there is no direct way to evaluate the performance of the clustering algorithms.

The structure of this paper is as follows. Different classification algorithms are presented in the next section. In Section 3 we discuss our data set and how the data was processed. Various evaluation techniques are described in Section 4. The results are given in Section 5. In Section 6 we present discussion and conclusions.

2 Classification algorithms

In this section we briefly introduce seven hierarchical clustering algorithms, k-means and neighbor-joining algorithm, originally used for reconstructing phylogenetic trees.

2.1 Hierarchical clustering

Cluster analysis is the process of partitioning a set of objects into groups or clusters (Manning and Schütze, 1999). The goal of clustering is to find structure in the data by finding objects that are similar enough to be put in the same group and by identifying distinctions between the groups. Hierarchical clustering algorithms produce a set of nested partitions of the data by finding successive clusters using previously established clusters. This kind of hierarchy is represented with a dendrogram—a tree in which more similar elements are grouped together. In this study seven hierarchical clustering algorithms will be investigated with regard to their performance on dialect pronunciation data. All these agglomerative clustering algorithms proceed from a distance matrix, repeatedly choosing the two closest elements and fusing them. They differ in the way in which distances are recalculated from the newly fused elements to the others. We now review the various calculations.

Single link method, also known as nearest neighbor, is one of the oldest methods in cluster analysis. The similarity between two clusters is computed as the distance

between the two most similar objects in the two clusters.

$$d_{k[ij]} = minimum(d_{ki}, d_{kj})$$

In this formula, as well as in other formulae in this subsection, i and j are the two closest points that have just been fused into one cluster$[i, j]$, and k represents all the remaining points (clusters). As noted in Jain and Dubes (1988), single link clusters easily chain together, producing the so-called *chaining effect*, and produce elongated clusters. The presence of only one intermediate object between two compact clusters is enough to turn them into a single cluster.

Complete link, also called furthest neighbor, uses the most distant pair of objects while fusing two clusters. It repeatedly merges clusters whose most distant elements are closest.

$$d_{k[ij]} = maximum(d_{ki}, d_{kj})$$

Unweighted Pair Group Method using Arithmetic Averages (UPGMA) belongs to a group of average clustering methods, together with three methods that will be described below. In UPGMA, the distance between any two clusters is the average of distances between the members of the two clusters being compared. The average is weighted naturally, according to size.

$$d_{k[ij]} = (n_i/(n_i + n_j)) \times d_{ki} + (n_j/(n_i + n_j)) \times d_{kj}$$

As a consequence, smaller clusters will be weighted less and larger ones more.

Weighted Pair Group Method using Arithmetic Averages (WPGMA), just as UPGMA, calculates the distance between the two clusters as the average of distances between all members of two clusters. But in WPGMA, the clusters that fuse receive equal weight regardless of the number of members in each cluster.

$$d_{k[ij]} = (\frac{1}{2} \times d_{ki}) + (\frac{1}{2} \times d_{kj})$$

Because all clusters receive equal weights, objects in smaller clusters are more heavily weighted than those in the big clusters.

Unweighted Pair Group Method using Centroids (UPGMC) In this method, the members of a cluster are represented by their middle point, the so-called centroid. This centroid represents the cluster while calculating the distance between the clusters to be fused.

$$d_{k[ij]} = (n_i/(n_i + n_j)) \times d_{ki} + (n_j/(n_i + n_j)) \times d_{kj} -$$

$$((n_i \times n_j)/(n_i + n_j)^2) \times d_{ij}$$

In the unweighted version of the centroid clustering the clusters are weighted based on the number of elements that belong to that cluster. This means that bigger clusters receive more weight, so that centroids can be biased towards bigger clusters. Centroid clustering methods can also occasionally produce reversals—partitions where the distance between two clusters being joined is smaller than the distance between some of their subclusters (Legendre and Legendre, 1998).

Weighted Pair Group Method using Centroids (WPGMC) Just as in WPGMA, in WPGMC all clusters are assigned the same weight regardless of the number of objects in each cluster. In that way the centroids are not biased towards larger clusters.

$$d_{k[ij]} = (\frac{1}{2} \times d_{ki}) + (\frac{1}{2} \times d_{kj}) - (\frac{1}{4} \times d_{ij})$$

Ward's method This method is also known as the minimal variance method. At each stage in the analysis clusters that merge are those that result in the smallest increase in the sum of the squared distances of each individual from the mean of its cluster.

$$d_{k[ij]} = ((n_k + n_i)/(n_k + n_i + n_j)) \times d_{ki} + ((n_k + n_j)/(n_k + n_i + n_j)) \times d_{kj} -$$

$$((n_k/(n_k + n_i + n_j)) \times d_{ij}$$

This method uses an analysis of variance approach to calculate the distances between clusters. It tends to create clusters of the same size (Legendre and Legendre, 1998).

2.2 K-means

The k-means algorithm belongs to the non-hierarchical algorithms which are often referred to as *partitional* clustering methods (Jain and Dubes, 1988). Unlike hierarchical clustering algorithms, partitional clustering methods generate a single partition of the data. A partition implies a division of the data in such a way that each instance can belong only to one cluster. The number of groups in which the data should be partitioned is usually determined by the user.

The k-means is the most commonly used partitional algorithm, that despite its simplicity, works sufficiently well in many applications (Manning and Schütze, 1999). The main idea of k-clustering is to find the partition of n objects into K clusters such that the total error sum of squares is minimized. In the most simple version, the algorithm consists of the following steps:

1. pick at random initial cluster centers
2. assign objects to the cluster whose mean is closest
3. recompute the means of clusters
4. reassign every object to the cluster whose mean is closest
5. repeat steps 3 and 4 until there are no changes in the cluster membership of any object

Two main drawbacks of the k-means algorithm are the following:

- the user has to define the number of clusters in advance
- the final partitioning depends on the initial position of the centroids

Possible solutions to these problems, as well as the detailed descriptions of the k-means algorithm can be found in some of the classical references to k-means: Hartigan (1975), Everitt (1980) and Jain and Dubes (1988).

2.3 Neighbor-joining

Apart from the seven hierarchical clustering algorithms and k-means, we also investigate the performance of the neighbor-joining algorithm. We introduce this technique at more length as it is less familiar to linguists. Neighbor-joining is a method for reconstructing phylogenetic trees that was first introduced by Saitou and Nei (1987). The main principle of this method is to find pairs of taxonomic units that minimize the total branch length at each stage of clustering. The distances between each pair of instances (in our case data collection sites) are calculated and put into the $n \times n$ matrix, where n represents the number of instances. The matrices are symmetrical since distances are symmetrical, i.e. distance (a, b) is always the same as distance (b, a). Based on the input distances, the algorithm finds a tree that fits the observed distances as closely as possible. While choosing the two nodes to fuse, the algorithm always takes into account the distance from every node to all other nodes in order to find the smallest tree that would explain the data. Once found, two optimal nodes are fused and replaced by a new node. The distance between the new node and all other nodes is recalculated, and the whole process is repeated until there are no more nodes left to be paired. The algorithm was modified by Studier and Kepler (1988), and the complexity was reduced to O(n^3). The steps of the algorithm are as follows (taken from Felsenstein (2004)):

- For each node compute u_i which is the sum of the distances from that node to all other nodes

$$u_i = \sum_{j:j \neq i}^{n} \frac{D_{ij}}{(n-2)}$$

- Choose i and j for which $D_{ij} - u_i - u_j$ is smallest

- Join i and j. Compute the length from i and j to the newly formed node v using the equations below. Note that the distances from the new node to its children (leaves) need not be identical. This possibility does not exist in hierarchical clustering.

$$v_i = \frac{1}{2} D_{ij} + \frac{1}{2}(u_i - u_j)$$

$$v_j = \frac{1}{2} D_{ij} + \frac{1}{2}(u_j - u_i)$$

- Compute the distance between the new node and all of the remaining nodes

$$D_{(ij),k} = \frac{(D_{ik} + D_{jk} - D_{ij})}{2}$$

- Delete nodes i and j and replace them by the new node

This algorithm produces a unique unrooted tree under the principal of minimal evolution (Saitou and Nei, 1987). In biology, the neighbor-joining algorithm has become very popular and widely used method for reconstructing trees from distance data. It is fast and can be easily applied to a large amount of data. Unlike most hierarchical clustering algorithms, it will recover the true tree even if there is not a constant rate of change among the taxa (Felsenstein, 2004).

3 Data preprocessing

The data set used in this research consists of transcriptions of the pronunciations of 156 words collected from 197 sites equally distributed all over Bulgaria. All measurements were done based on the phonetic distances between the various pronunciations of these 156 words. No morphological, lexical or syntactic variation between the dialects were taken into account.

Word transcriptions were preprocessed in the following way:

- First, all diacritics and suprasegmentals were removed from word transcriptions. In order to process diacritics and suprasegmentals, they should be assigned certain weights appropriate for the specific language that is being analyzed. Since no study of this kind was available for Bulgarian, diacritics and suprasegmentals were removed, which resulted in the simplification of data representation. For example, [u], [u:], [ˈu], and [ˈuː] counted as the same phone. Also, all words were represented as series of phones which are not further defined. The result of comparing two phones can be 1 or 0; they either match or they do not. For example, pair [e, ɛ] counts as different to the same degree as pair [e, i]. Although it is linguistically counterintuitive to use less sensitive measures, Heeringa (2004:p.186)

has shown that in the aggregate analysis of dialect differences more detailed feature representation of segments does not improve the results obtained by using simple phone representation.

- All transcriptions were aligned based on the following principles: a) a vowel can match only with a vowel b) a consonant can match only with a consonant, semivowels [j], [w] and sonorants. The alignments were carried out using the Levenshtein algorithm, which also results in the calculation of a distance between each pair of words. A detailed explanation of the Levenshtein algorithm can be found in Heeringa (2004). The distance is the smallest number of insertions, deletions, and substitutions needed to transform one string to the other. In this work all three operations were assigned the same value: 1. An example of an aligned pair of transcriptions can be seen here:

-	e	d	e	m
j	ɑ	d	ɑ	-

The distance between two sites is the mean of all word distances calculated for those two sites. The final result is a distance matrix which contains the distances between each two sites in the data set. This distance matrix was further analyzed using seven hierarchical algorithms, k-means and the neighbor-joining algorithm described in the previous section.

4 Evaluation

We analyzed the results obtained by the above mentioned methods further using a variety of measures. Multidimensional scaling was performed in order to see if there were any separate groups in the data and to determine the optimal number of clusters in the data set. External validation of the clustering results included the modified Rand index, purity and entropy. External validation involves comparison of the structure obtained by different algorithms to a *gold standard*. In our study we used the manual classifica-

tion of all the sites produced by traditional dialectologist as a *gold standard*. Internal validation included examining the cophenetic correlation coefficient, noisy clustering and a consensus tree, which do not require comparison to any *a priori* structure, but rather try to determine if the structure obtained by algorithms is intrinsically appropriate for the data.

Multidimensional scaling is a dimension-reducing method used in exploratory data analysis and a data visualization method, often used to look for separation of the clusters (Legendre and Legendre, 1998). The goal of the analysis is to detect meaningful underlying dimensions that allow the researcher to explain observed similarities or dissimilarities between the investigated objects. In general then, MDS attempts to arrange "objects" in a space with a certain small number of dimensions, which, however, accord with the observed distances. As a result, we can “explain“ the distances in terms of underlying dimensions. It has been frequently used in linguistics and dialectology since Black (1973).

4.1 External validation

The **modified Rand index** (Hubert and Arabie, 1985) is used for comparing two different partitions of a finite set of objects. It is a modified form of the Rand index (Rand, 1971), one of the most popular measures for comparing partitions. Given a set of n elements $S = o_1, ...o_n$ and two partitions of S, $U = u_1, ...u_R$ and $V = v_1, ...v_C$ we define

a the number of pairs of elements in S that are in the same set in U and in the same set in V

b the number of pairs of elements in S that are in different sets in U and in different sets in V

c the number of pairs of elements in S that are in the same set in U and in different sets in V

d the number of pairs of elements in S that are in different sets in U and in the same set in V

The Rand index R is

$$R = \frac{a+b}{a+b+c+d}$$

In this formula a and b are the number of pairs of elements in which two classifications

agree, while c and d are the number of pairs of elements in which they disagree. The value of the Rand index is between 0 and 1, with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same. In dialectometry, this index was used by Heeringa et al. (2002) to validate dialect comparison methods. A problem with the Rand index is that it does not return a constant value (zero) if two partitions are picked at random. Hubert and Arabie (1985) suggested a modification of Rand index that corrects this property. It can be expressed in the general form as:

$$\frac{RandIndex - ExpectedIndex}{MaximumIndex - ExpectedIndex}$$

The value of the modified Rand index is between -1 and 1.

Entropy and **purity** are two measures used to evaluate the quality of clustering by looking at the reference class labels of the elements assigned to each cluster (Zhao and Karypis, 2001). Entropy measures how different classes of elements are distributed within each cluster. The entropy of a single cluster is calculated using the following formula:

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r}$$

where S_r is a particular cluster of size n_r, q is the number of classes in the reference data set, and n_r^i is the number of the elements of the ith class that were assigned to the rth cluster. The overall entropy is the sum of all cluster entropies weighted by the size of the cluster:

$$E = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r)$$

The **purity** measure is used to determine to which extent a cluster contains objects from primarily one class. The purity of a cluster is calculated as:

$$P(S_r) = \frac{1}{n_r} max(n_r^i)$$

while the overall purity is the weighted sum of the individual cluster purities:

$$P = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r)$$

4.2 Internal validation

The **cophenetic correlation coefficient** (Sokal and Rohlf, 1962) is Pearson's correlation coefficient computed between the cophenetic distances produced by clustering and those in the original distance matrix. The cophenetic distance between two objects is the similarity level at which those two objects become members of the same cluster during the course of clustering (Jain and Dubes, 1988) and is represented as branch length in dendrogram. It measures to which extent the clustering results correspond to the original distances. When the clustering functions perfectly, the value of the cophenetic correlation coefficient is 1. In order to check the significance of this statistics we performed the simple Mantel test as implemented in **zt** software (Bonet and de Peer, 2002). A simple Mantel test is used to compare two matrices by testing the correlation between them using the standard Pearson correlation coefficient and testing its statistical significance (Mantel, 1967).

Noisy clustering, also called composite clustering, is a procedure in which small amounts of random noise are added to matrices during repeated clustering. The main purpose of this procedure is to reduce the influence of outliers on the regular clusters and to identify stable clusters. As shown in Nerbonne et al. (2008) it gives results that nearly perfectly correlate with the results obtained by bootstrapping—a statistical method for measuring the support of a given edge in a tree (Felsenstein, 2004). The advantage of the noisy clustering, compared to bootstrapping, is that it can be applied on a single distance matrix—the same one used as input for the classification algorithms.

A **consensus dendrogram**, or consensus tree, is a tree that summarizes the agreement between a set of trees (Felsenstein, 2004). A consensus tree that contains a large number of internal nodes shows high agreement between the input trees. On the other hand, if a consensus tree contains few internal nodes, it is a sign that input trees classify the data in conflicting ways. The majority rule consensus tree, used in this study,

is a tree that consists of the groups, i.e clusters, which are present in the majority of the trees under study. In this research a consensus dendrogram was created from four dendrograms produced by four different hierarchical clustering methods. Clusters that appear in the consensus tree are those supported by the majority of algorithms and can be taken with greater confidence to be true clusters.

5 Results

Before describing the results of applying various algorithms to our data set, we give a short description of the traditional division of the Bulgarian dialect area that we used for external validation in our research.

5.1 Traditional scholarship

Traditional scholarship (Stojkov, 2002) divides the Bulgarian language into two main groups: Western and Eastern. The border between these two areas is so-called 'yat' border that reflects different pronunciations of the old Slavic vowel 'yat'. It goes from Nikopol in the North, near Pleven and Teteven down to Petrich in the South (bold dashed line in Figure 1).

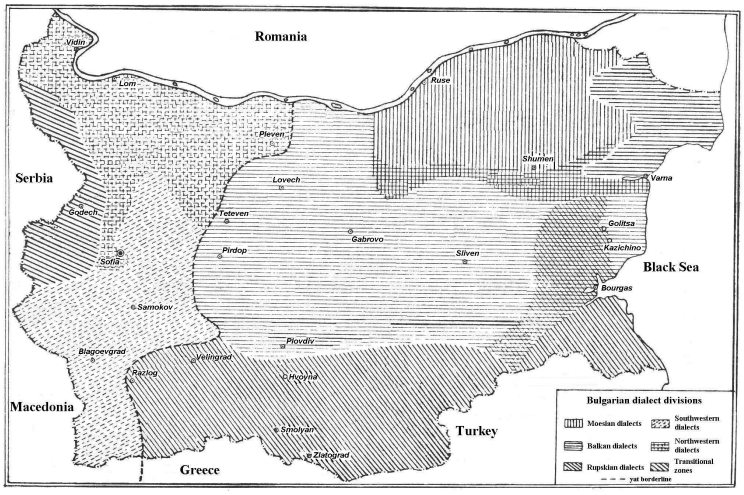

Figure 1: Traditional map of Bulgarian dialects

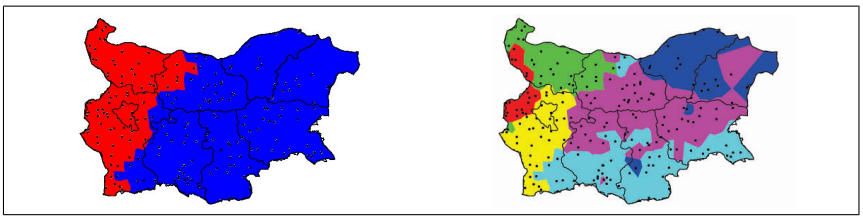

Figure 2: The two-way and six-way classification of sites done by expert

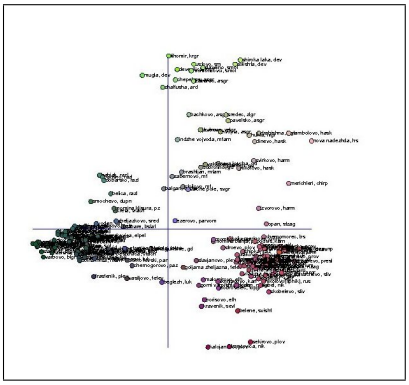

Figure 3: MDS plot

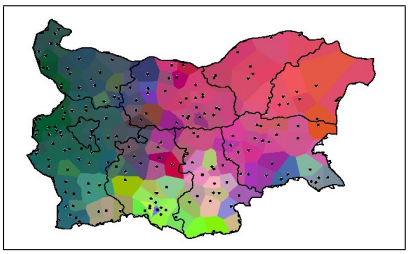

Figure 4: MDS map

Stojkov divides each of these two areas further into three smaller dialect zones, which can also be seen on the map in Figure 1. This 6-fold division is based on the variation of different phonetic features. No morphological or syntactic differences were taken into account. In order to evaluate the performance of different clustering algorithms, all sites present in our data set were manually put by an expert in one of the two, and later into six, main dialect areas according to the Stojkov's classification. This was done by Professor Vladimir Zhobov, phonetician and dialectologist from the Faculty of Slavic Philologies 'St. Kliment Ohridski', University of Sofia.

Due to various historical events, mostly migrations, some villages are dialectological islands surrounded by language varieties from groups different from the one they belong to. This lack of geographical coherence can be seen, for example, in the north-central part on the map in Figure 2.

5.2 MDS

Multidimensional scaling was performed in order to check if there are any separate clusters in the data. The results can be seen in the Figure 3, where the first two extracted dimensions are plotted against the x and y axes. In addition, all three extracted dimensions are represented by different shades of red, blue and green colors. This represents the third MDS dimension.

The first three dimensions represented in Figure 3 explain 98 per cent of the variation in the data set—the first dimension extracted explains 80 per cent of the variation, and the second dimension 16 per cent. In Figure 3 we can see two distinct clusters along the x-axis, which, if put on the map, correspond to the Eastern and Western group of dialects (Figure 4).

Variation along the y-axis corresponds to the separation of the dialects in the South from the rest of the country. Using MDS to screen the data, we observe that there are two distinct clusters in the data set—even though MDS is fully capable of representing continuous data. This finding fully agrees with the expert opinion (Stojkov, 2002) according to which the Bulgarian dialect area can be divided into Eastern and Western dialect areas along the 'yat' border. A third area that can be seen in Figure 4 is the area

in the South of the country—the area of the Rodopi mountains. In the classification of dialects done by Stojkov (2002), this area is identified as one of the six main dialect areas based on the phonetic features.

5.3 External validation

The results of the multidimensional scaling and dialect divisions done by expert can be used as a first step in the evaluation of the clustering algorithms. Visual inspection shows that three algorithms fail to identify any structure in the data, including East-West division of the dialects: single link and two centroid algorithms, UPGMC and WPGMC. Dendrograms drawn using UPGMC and WPGMC reveal a large number of reversals, while closer inspection of the single link dendrogram clearly shows the presence of the *chain effect*. The remaining algorithms reveal the East-West division of the country clearly (Figure 5). For that reason, in the rest of the paper the main focus will be on those four clustering algorithms, as well as on the k-means and neighbor-joining.

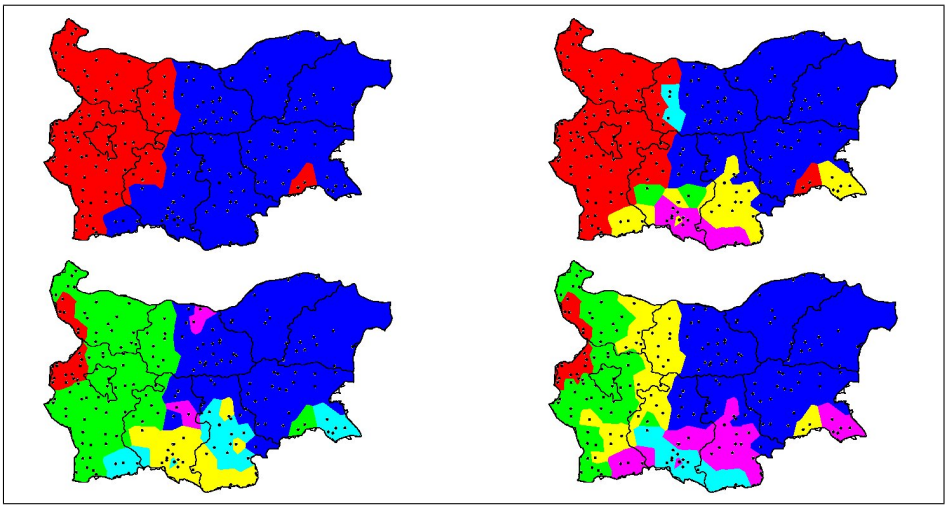

Figure 5: Top left map: 2-way division produced by UPGMA, WPGMA and Ward's method. Top right map: 6-way division produced by UPGMA. Bottom maps: 6-way divisions produced by WPGMA and Ward's method respectively.

In order to compare divisions done by clustering algorithms with the division of sites done by expert we calculated the modified Rand index, entropy and purity for

Table 1: Results of external validation: the modified Rand index (MRI), entropy (E) and purity (P). Results for the 2, 3 and 6-fold divisions are reported.

Algorithm	MRI(2)	MRI(3)	MRI(6)	E(2)	E(3)	E(6)	P(2)	P(3)	P(6)
single link	-0.004	0.007	-0.001	0.958	0.967	0.881	0.614	0.396	0.360
complete link	0.495	0.520	0.350	0.510	0.542	0.467	0.848	0.766	0.645
UPGMA	**0.700**	**0.627**	0.273	0.368	**0.445**	0.583	0.914	**0.853**	0.568
WPGMA	**0.700**	0.626	0.381	0.368	0.445	0.448	0.914	**0.853**	0.665
UPGMC	-0.004	0.007	-0.006	0.959	0.967	0.926	0.614	0.396	0.310
WPGMC	-0.004	0.007	-0.005	0.958	0.967	0.925	0.614	0.396	0.305
Ward's method	**0.700**	**0.627**	0.398	0.368	**0.445**	0.441	0.914	**0.853**	0.675
k-means	**0.700**	0.625	**0.471**	**0.354**	0.451	**0.355**	**0.919**	0.756	**0.772**
NJ	0.567	0.461	-	0.442	0.550	-	0.873	0.777	-

the 2-fold, 3-fold, and 6-fold divisions done by algorithms on the one hand, and those divisions according to the expert on the other. The results can be seen in Table 1.

The neighbor-joining algorithm produced an unrooted tree (Figure 6), where only 2-fold and 3-fold divisions of the sites can be identified. Hence, all the indices were calculated only for the 2-fold and 3-fold divisions in neighbor-joining.

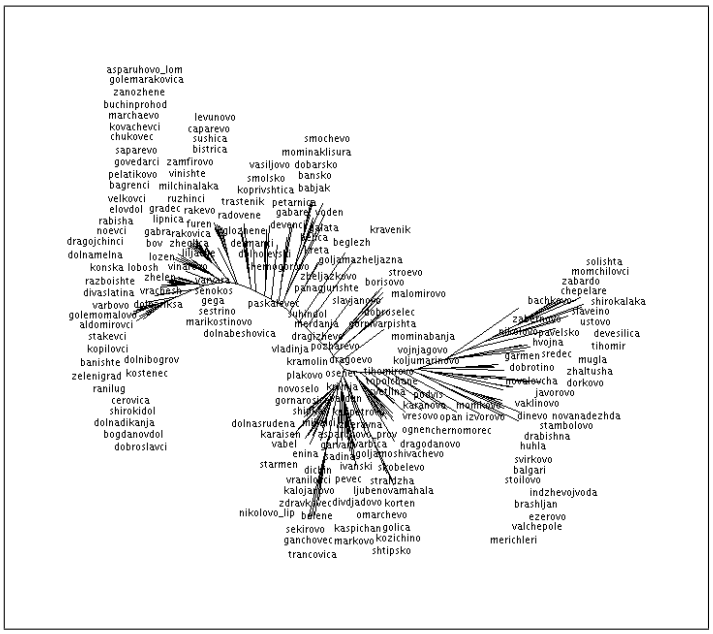

Figure 6: NJ tree

In Table 1 we can see that the values of the modified Rand index for single link and two centroid methods are very close to 0, which is the value we would get if the partitions were picked at random. UPGMA, WPGMA, Ward's method and k-means, which gave

nearly the same 2-fold division of the sites, show the highest correspondences with the divisions done by expert. For 3-fold and 6-fold divisions the values for the modified Rand index went down for all algorithms, which was expected since the number of groups increased. The two algorithms with the highest values of the index are Ward's method and UPGMA for 3-fold, and k-means for the 6-fold division. Just as in the case of the 2-fold division, the single-link, UPGMC, and WPGMC algorithms have values of the modified Rand index close to 0. Neighbor-joining produced a relatively low correspondence with expert opinion for the 3-fold division—0.461. Similar results for all algorithms and all divisions were obtained using entropy and purity measures. External validation of the clustering algorithms has revealed that single link, UPGMC and WPGMC algorithms are not suitable for the analysis of the data we are working with, since they fail to recognize any structure in the data.

5.4 Internal validation

In the next step internal validation methods were used to check the performance of the algorithms: the cophenetic correlation coefficient, noisy clustering and consensus tree. Since k-means does not produce a dendrogram, it was not possible to calculate the cophenetic correlation coefficient. The values of the cophenetic correlation coefficient for the remaining eight algorithms can be seen in Table 2. We can see that clustering results of the UPGMA have the highest correspondence to the original distances of all algorithms—90.26 per cent. They are followed by the results obtained by using complete link and neighbor-joining algorithm. All correlations are highly significant with $p < 0.0001$. Given the poor performance of the centroid and single-link methods in detecting the dialect divisions scholars agree on, we note that cophenetic correlation coefficients are not successful in distinguishing the better techniques from the weaker ones. We conjecture that the reason for this lies in the fact that the cophenetic correlation coefficient so dependent is on the lengths of the branches in the dendrogram, while our primary purpose is the classification.

Noisy clustering, that was applied with the seven hierarchical algorithms, has confirmed that there are two relatively stable groups in the data: Eastern and Western.

Table 2: Cophenetic correlation coefficient

Algorithm	CCC	p
single link	0.7804	0.0001
complete link	0.8661	0.0001
UPGMA	**0.9026**	0.0001
WPGMA	0.8563	0.0001
UPGMC	0.8034	0.0001
WPGMC	0.6306	0.0001
Ward's method	0.7811	0.0001
Neighbor-joining	0.8587	0.0001

Dendrograms obtained by applying noisy clustering to the whole data set show low confidence for the two-way split of the data, between 52 and 60 per cent. After removing the Southern villages from the data set, we obtained dendrograms that confirm two-way split of the data along the 'yat' border with much higher confidence ranging around 70 per cent. These values are not very high. In order to check the reason of the influence of the Southern varieties on the noisy clustering we examine an MDS plot in two dimensions with cluster groups marked by colours. In Figure 7 we can see MDS plot of 6 groups produced by WPGMA algorithm. MDS plot reveals two homogeneous groups and a third, more diffuse, group that lies at a remove from them. The third group of the sites represents the Southern group of varieties, colored light blue and yellow, and is much more heterogeneous than the rest of the data. Closer inspection of the MDS plot in Figure 3 also shows that this group of dialects has a particularly unclear border to the Eastern dialects, which could explain the results of the noisy clustering applied to the whole data set.

Since different algorithms gave different divisions of sites, we used a consensus dendrogram in order to detect the clusters on which most algorithms agree. Since single link, UPGMC and WPGMC have turned to be inappropriate for the analysis of our data, they were not included in the consensus dendrogram. The consensus dendrogram drawn using complete link, UPGMA, WPGMA and Ward's method can be seen in Figure 8. The names of the sites are colored based on the expert's opinion, i.e. the same as in Figure 2. The dendrogram shows strong support for the East-West division of sites, but no agreement on the division of sites within the Eastern and Western areas.

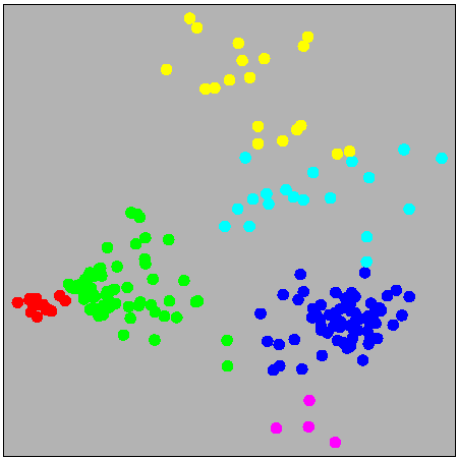

Figure 7: MDS plot of 6 clusters produced by WPGMA. Note that the good separation of the clusters is often spoiled by unclear margins.

At this level of hierarchy, i.e. 2-way division, there are several sites classified differently by algorithms and by expert. These sites go along the 'yat' border and represent the marginal cases. The only two exceptions are villages in the South-East, namely Voden and Zheljazkovo. However, according to many traditional dialectologists these villages should be classified as Western dialects due to many features that they share with the dialects in the West (personal communication with prof. Vladimir Zhobov). The four algorithms show agreement only on the very low level where several sites are grouped together and on the highest level. It is not possible to extract any hierarchical structure that would be present in the majority of four analyses.

6 Discussion and conclusions

Different clustering validation methods have shown that three algorithms are not suitable at all for the data we are working with, namely single link, UPGMC and WPGMC. The remaining four hierarchical clustering algorithms gave different results depending on the level of hierarchy, but all four algorithms had high agreement on the detection of two main dialect areas within the dialect space. At the lower level of hierarchy, i.e where there are more clusters, the performance of the algorithms is poorer, both with

respect to the expert opinion and with respect to the mutual agreement as well. As shown by noisy clustering, the 2-fold division of the Bulgarian language area is the only partition of sites that can be asserted with high confidence.

The results of the neighbor-joining algorithm were a bit less satisfactory. The reason for this could be in the fact that our data is not tree-like, but rather contains a lot of borrowings due to contact between different dialects. A recent study (Hamed and Wang, 2006) of Chinese dialects has shown that their development is not tree-like and that in such cases usage of tree-reconstruction methods can be misleading.

The division of sites done by the k-means algorithm corresponded well with the expert divisions. Two and three-way divisions also correspond well with the divisions of four hierarchical clustering algorithms. What we find more important is the fact that in the divisions obtained by the k-means algorithm into 2, 3, 4, 5 and 6 groups the two-way division into the Eastern and Western groups is the only stable division that appears in all partitions.

This research shows that clustering algorithms should be applied with caution as classifiers of language dialect varieties. Where possible, several internal and external validation methods should be used together with the clustering algorithms in order to validate their results and make sure that the classifications obtained are not mere artifacts of algorithms but natural groups present in the data set. Since performance of clustering algorithms depends on the sort of data used, evaluation of algorithms is a necessary step in order to obtain results that can be asserted with high confidence.

The fact that there are only two distinct groups in our data set that can be asserted with high confidence, as opposed to six found in the traditional atlases, could possible be due to the simplified representation of the data (see Section 3). It is also possible that some of the features responsible for the traditional 6-way division are not present in our data set. At the moment, we are investigating these two issues. Regardless of the quality of the input data set, we have shown that clustering algorithms will partition data into the desired number of groups even if there is no natural separation of the data. For this reason it is essential to use different evaluation techniques along with the clustering algorithms.

Classification algorithms are nowadays applied in different subfields of humanities (Woods et al., 1986; Boonstra et al., 1990). It is a general technique that can be applied to any sort of data that needs to be put into different groups in order to discover various patterns. Document and text classification, authorship detection and language typology are just some of the areas where classification algorithms are nowdays successfully applied. The problem of choosing the right classification algorithm and obtaining stable results goes beyond dialectometry and is present whenever applied. For this reason the present paper is valuable not only for the research done in dialectometry, but also for other branches of humanities that are using clustering techniques. It shows how unstable the results of clustering algorithms can be, but also how to approach this problem and overcome it.

References

P. Black (1973), 'Multidimensional scaling applied to linguistic relationships', in *Cahiers de l'Institut de Linguistique Louvain*, Volume 3 (Montreal). Expanded version of a paper presented at the Conference on Lexicostatistics. University of Montreal.

E. Bonet and Y. V. de Peer (2002), 'zt: a software tool for simple and partial mantel tests', *Journal of Statistical software*, 7(10), 1–12.

O. Boonstra, P. Doorn, and F. Hendrickx (1990), *Voortgezette Statistiek voor Historici* (Muiderberg).

B. S. Everitt (1980), *Cluster Analysis* (New York).

J. Felsenstein (2004), *Inferring Phylogenies* (Massachusetts).

H. Goebl (2007), 'On the geolinguistic change in Northern France between 1300 and 1900: a dialectometrical inquiry', in J. Nerbonne, T. M. Ellison, and G. Kondrak, eds, *Computing and Historical Phonology. Proceedings of the Ninth Meeting of the ACL Special Interest Group in Computational Morphology and Phonology* (Prague), 75–83.

M. B. Hamed and F. Wang (2006), 'Stuck in the forest: Trees, networks and Chinese dialects', *Diachronica*, 23(1), 29–60.

J. A. Hartigan (1975), *Cluster algorithms* (New York).

W. Heeringa (2004), *Measuring Dialect Pronunciation Differences using Levensthein Distance* (PhD Thesis, University of Groningen).

W. Heeringa, J. Nerbonne, and P. Kleiweg (2002), 'Validating dialect comparison methods', in W. Gaul and G. Ritter, eds, *Classification, Automation, and New Media. Proceedings of the 24th Annual Conference of the Gesellschaft für Klassifikation, University of Passau, March 15-17, 2000* (Heidelberg), 445–452.

L. Hubert and P. Arabie (1985), 'Comparing partitions', *Journal of Classification*, 2, 193–218.

A. K. Jain and R. C. Dubes (1988), *Algorithms for Clustering Data* (New Yersey).

P. Legendre and L. Legendre (1998), *Numerical Ecology*, second ed. (Amsterdam).

C. Manning and H. Schütze (1999), *Foundations of Statistical Natural Language Processing* (Cambridge, MA).

N. Mantel (1967), 'The detection of disease clustering and a generalized regression approach', *Cancer Research*, 27, 209–220.

H. Moisl and V. Jones (2005), 'Cluster analysis of the Newcastle Electronic Corpus of Tyneside English: a comparison of methods', *Literary and Linguistic Computing*, 20, 125–146.

J. Nerbonne, P. Kleiweg, W. Heeringa, and F. Manni (2008), ' Projecting Dialect Differences to Geography: Bootstrap Clustering vs. Noisy Clustering', in H. B. Christine Preisach, Lars Schmidt-Thieme and R. Decker, eds, *Data Analysis, Machine Learning, and Applications. Proc. of the 31st Annual Meeting of the German Classification Society* (Berlin), 647–654.

J. Nerbonne and C. Siedle (2005), 'Dialektklassifikation auf der Grundlage Aggregierter Ausspracheunterschiede', *Zeitschrift für Dialektologie und Linguistik*, 72(2), 129–147.

W. M. Rand (1971), 'Objective criteria for the evaluation of clustering methods', *Journal of American Statistical Association*, 66(336), 846–850.

N. Saitou and M. Nei (1987), 'The neighbor-joining method: A new method for reconstructing phylogenetic trees', *Molecular Biology and Evolution*, 4, 406–425.

R. R. Sokal and F. J. Rohlf (1962), 'The comparison of dendrograms by objective methods', *Taxon*, 11, 33–40.

S. Stojkov (2002), *Bulgarska dialektologiya* (Sofia).

J. A. Studier and K. J. Kepler (1988), 'A note on the neighbor-joining algorithm of Saitou and Nei', *Molecular Biology and Evolution*, 5, 729–731.

A. Woods, P. Fletcher, and A. Hughes (1986), *Statistics in Language Studies* (Cambridge).

Y. Zhao and G. Karypis (2001), 'Criterion functions for document clustering: Experiments and analysis', Technical report 01-40, Department of Computer Science, University of Minnesita, Minneapolis, MN.

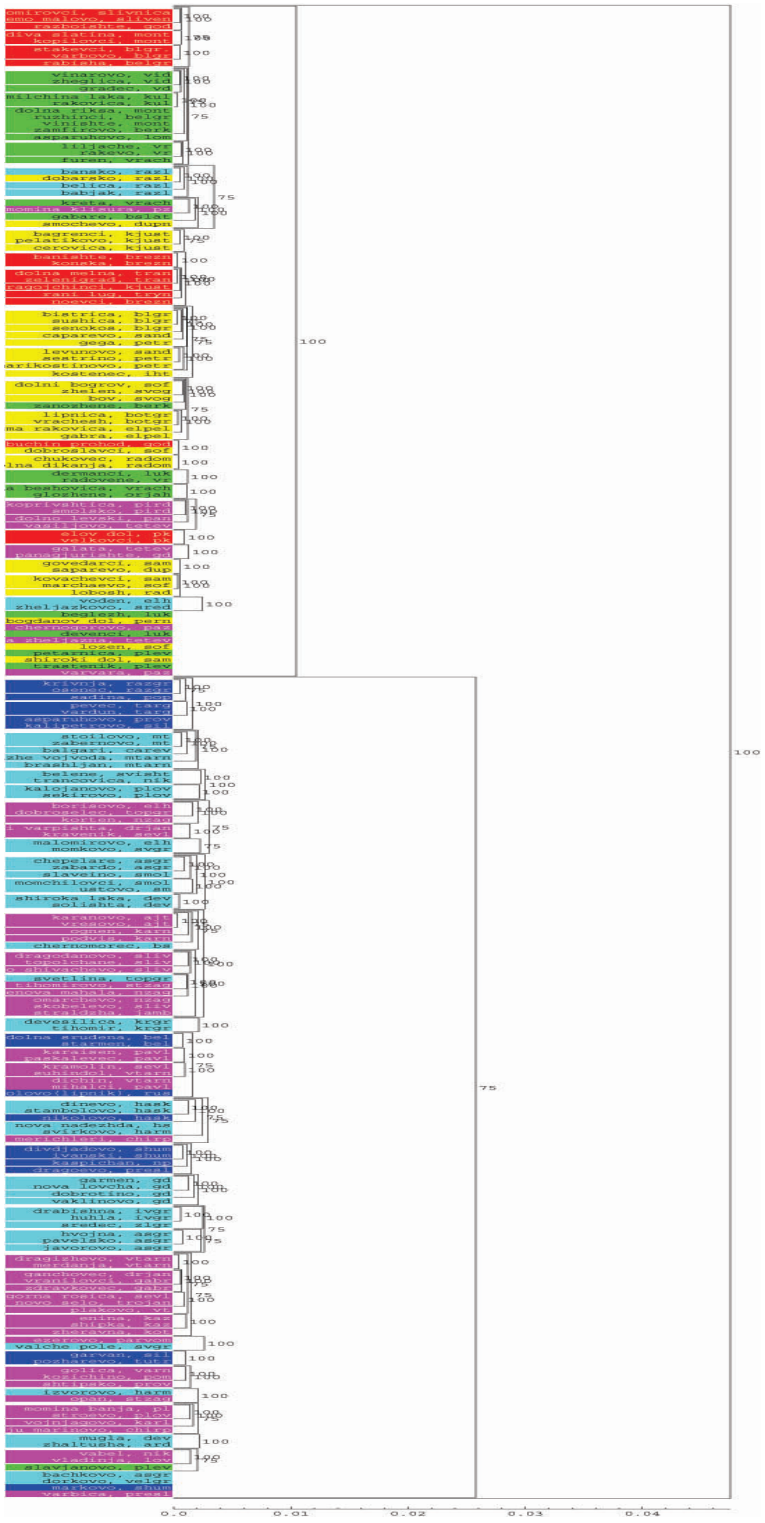

Figure 8: Consensus dendrogram for four algorithms.The four algorithms show agreement only on the 2-way division. It is not possible to extract any hierarchical structure that would be present in the majority of four analyses. (For the explanation of colors see Figure 3.)