Comparison of linguistic language classification based on origin and data driven language classification using the IPA and clustering

Isa Rethans[1] , **Marco Loog**[1] , **Tom Viering** [1] , **Stavros Makrodimitris**[1] , **Arman Naseri Jahfari**[1]

[1]TU Delft

Abstract

Language similarity is very useful for enrichment data in both Natural Lanuguage Processing (NLP) and Automatic Speech Recognition (ASR). A clustering algorithm could provide an efficient means to define language similarity in a data-driven way. This research investigates the relation between linguistic classification by origin and data driven classification based on the pronunciation of languages using k-means clustering where the focus is placed on the Indo-European languages. The results show large variation in cluster results and consequently large variation in correspondence with linguistic classification. This is caused by a relatively even spread of the data over the feature space. Still, the results indicate significance in the relation between the two classification methods. Furthermore, this research functions as a foundation and a source of inspiration for a lot of possible future research.

Natural Language Processing (NLP) techniques are becoming increasingly sophisticated. In a similar way, Automatic Speech Recognition (ASR) methods experience rapid development. Both NLP and ASR are linguistic fields related to Computer Science. With many languages in the world a lot of great use can be achieved. For example, the combination of the two fields results in something that comes closest to real conversation between human and machine. However, for many languages there is little to no data available. Language similarity has proven to be very useful for enrichment of NLP and ASR data. One study used the similarity of the Russian and Czech language to create a resource light morphological tagging mechanism for Russian [11]. Additionally, the study by Xia et. al (2007) uses resource-rich languages to enrich data for similar languages without a lot of data available [26].

The field of linguistics defines two different language classifications. First of all, there is genetic classification which considers the relatedness based on history and origin of languages. Secondly, there is typological classification which is based on structural characteristics of language [18]. To form the language classification based on genetic or typological features, a lot of linguistic knowledge is required. A clustering algorithm could provide an efficient means to investigate the option of describing language similarity in a data-driven way. The characteristic that is used for clustering is pronunciation since it is very prominent in ASR and it is a simple feature that is available for every language. The International Phonetic Alphabet (IPA) describes pronunciation in written form. As this is used for the pronunciation of all languages, no difficulties will arise with varying alphabets.

Cluster evaluation is not straightforward due to absence of an objective measure for language similarity based on pronunciation. It is useful to have some reference to compare the classification to. This may be found in language origin. The fact that there is a relation between the origin of a language and its pronunciation gives reason to believe that there is a possibility for a relation between language classification based on origin and language classification based on pronunciation.

The main question of this research is: ”How does data-driven language classification using IPA and clustering compare to linguistic language classification based on origin?”. This research focuses on Indo-European languages. This allows to narrow down the research, while still being able to make a proper comparison. Furthermore, a selection of languages differing from the Indo-European Languages is used to verify that the similarity with those is very low.

This study uses Term Frequency - Inverse Document Frequency (TF-IDF) to create a numerical representation of the data per language. After this Singular Value Decomposition (SVD) is applied to reduce the dimension of the produced vector. This results in a useful form of data which is ready to be fed into a clustering algorithm. Then, k-means clustering is applied and evaluated using various methods to provide insights from different points of view.

The remainder of this paper is structured as follows. Chapter 2 provides the reader with the background knowledge of terms and techniques used. Chapter 3 goes into related work. Chapter 4 focuses on the experimental set up of the clustering, including data extraction, data processing and vectorisation. Chapter 5 contains the results and a immediate topic specific discussion. A more general discussing follows in Chapter 6. Chapter 7 provides ideas for future research and in Chapter 8 conclusions are drawn. Lastly, Chapter 9 addresses ethics and reproducibility.

1 Background information

In this paper a few domain specific terms, techniques and methods are used. This section addresses this to improve understanding and ease reading the next sections.

1.1 IPA

Since the aim is to try and find patterns in pronunciation, all the words in the dataset are written in IPA. This alphabet is specifically designed to have a representation for the spoken form and mostly uses characters from the latin alphabet. It aims to provide an unique symbol for each distinctive sound that exists in spoken language [6]. The idea behind this alphabet is to have one general representation of spoken language independent of the variation of alphabets between languages. This is very convenient for the clustering, because it allows to easily use and compare words from all languages without having to take notice of different alphabets.

1.2 N-grams

The words in the raw data undergo some processing steps, which will be described in more detail in the Method section. What matters here is that one of the steps is the creation of n-grams. N-grams are sequences of n symbols. Each pronunciation is mapped to all possible n-grams by applying a moving window on a word with window size n. Figure 1 gives an example.

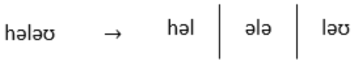


Figure 1: Example: 3-gram split for hello in IPA

1.3 Vectorization with TD-IDF

To be able to run a clustering algorithm the data must be in a way that a computer can comprehend. It is not possible to just have a lot of text as input. Therefore there is necessity for a numerical representation. TF-IDF is one of the most popular techniques in NLP to achieve that [23]. It is a term weighting technique that indicates the importance of a term within a document. It counts how often a term occurs in a document (term frequency) and scales it by a factor that represents how popular the term is in other documents (inverse document frequency). A high value implies a strong relation with the document. In this research a n-gram is a term and a language is a document.

1.4 Dimensionality Reduction

The goal of dimensionality reduction is to reduce the size of the feature space while preserving the most important features of the original data. [15]. Singular Value Decomposition (SVD) accomplices that by matrix factorization. It decomposes a matrix Y with dimensions $m \times p$ into USV^T, with dimensions $m \times k$, $k \times k$ and $p \times k$ respectively [2]. So at the start there are m languages and p features (n-grams) and after SVD only the k most relevant features remain. Dimensionality reduction is required once more for 2-dimensional visualisation. For this purpose t-SNE is very appropriate [2]. The resulting 2-dimensional vectors have a high probability to be close together when they are similar and a high probability to be relatively far away if they are not similar.

K-Means Clustering

The goal of clustering is to assign labels to a set of objects in a way that objects with the same label have more in common than objects with another label. The method used is k-means clustering, which is a partitional clustering method that aims to divide the data into k disjoint subsets [17]. The algorithm starts by randomly picking k initial centroids for each cluster. Next, each point is assigned to the nearest centroid. After which the centroids are reassigned to be the centre points of the current clustering. The last two steps are repeated until there is no further change [14].

1.5 Silhouette Score

The silhouette score is a a standard cluster evaluation method which indicates cluster coherence. It takes values between -1 and 1 [3]. Let l be an feature vector assigned to a cluster A. Also, we define $d(l, C)$ to be the average distance from l to all the other points in a cluster C. Then, $d(l, A)$ is the average distance of l to all the other vectors in its own cluster A, which we denote by $a(l)$. Next, we take the minimum $d(l, C)$ over all the clusters except A. We define B to be the cluster for which this minimum is reached and $b(l)$ to contain this minimum value ($d(l, C) = b(l)$). So, B is the closest cluster for l other than A. Now the silhouette score is:

$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

The interpretation of this value is eased by looking at its extremities. A value of $s(i)$ close to 1 means that the average distance within a cluster is much smaller than the average distance with the closest cluster. Therefore i has been assigned to an appropriate cluster. If $s(i)$ is close to 0, $a(i)$ and $b(i)$ are approximately the same. Hence there is much more indifference about the assignment to either A or B. Finally if $s(i)$ is close to -1, it means that l is on average closer to B then A, so this almost indicates classification. To calculate the silhouette score for the whole data set, the average is taken over all vectors. [24].

1.6 Rand-Index

The Rand-Index is a value indicating the correspondence between two distinct partitions of the same set. It takes into account all possible pairs of objects in a set. Let U and V be two partitions of a set S. Then two options arise: either U and V are in agreement or they are not. Being in agreement in this case, means that a pair of objects is either in the same partition in both U and V or in a different partition in both U and V. Whereas disagreement means that the objects are in the same partition in U and in a different partition in V or vice versa [12]. Now the rand index is:

$$RI = \frac{\textit{Count of Pairs in Agreement}}{\textit{Total Number of Pairs}}$$

A drawback is that this metric gives high values when comparing two random partitions. To account for chance the adjusted rand index (ARI) exists, which normalizes the Rand

Index using the expected value for the similarity of two random partitions. The ARI has a value between -1 and 1 and is 0 for the expected value of random cluster assignment [25].

1.7 Elbow method

The elbow method is a way to determine the optimal k in k-means clustering. The idea is to try increasing values for k and calculate the distortion. The distortion is defined as the sum of the squared distances between each observation and the corresponding cluster centre. A plot with k on the x-axis and distortion on the y-axis will result in a decreasing and convex curve like Figure 2. The optimal value for k is located at the inflection point. The intuition behind this is that the distortion declines rapidly when k approaches the actual number of clusters. Whereas, exceeding the actual number of clusters will result in decline at a decreasing rate [28].

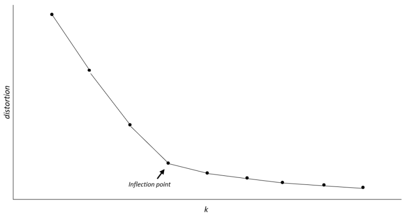


Figure 2: Example: 3-gram split for hello in IPA

2 Related work

A general way to see this research is like a text clustering problem. This problem occurs often and thus many research on this topic is available. Needless to say, each research has its own approach that fits the specific problem. In clustering, many design choices have to be made. Some of the most important are the way to define a representative numerical representation of the data, which clustering algorithm to choose and which parameters for that specific algorithm to use.

For term weighting, TF-IDF is a widely adopted approach [21]. However, it results in high dimensions which are not efficient to work with performance wise (computation time) and memory wise. Moreover, vectors in a high dimensional space appear equidistant to each other, this phenomenon is known as the curse of dimensionality [13]. The uniformity in the distances between all vectors make them appear equally alike, which complicates forming meaningful clusters. Therefore, TF-IDF is often combined with a dimensionality reduction technique. This reduces the vector dimensions enormously with higher clustering accuracy, speed up and better topic matching. [15]. Many variants exits. As SVD is designed for decomposition of matrices, it is very appropriate for the the TF-IDF matrix. For machine learning tasks regarding texts, Latent Semantic Analysis (LSA) is most often used. This method is based on SVD and is aimed to discover semantic relations between features [2]. However, semantic relations do not contribute to sound similarity. Therefore SVD seems the more appropriate option.

Lastly, other comparisons of data driven and genetic language classification exist. However, these studies all use language characteristics other than pronunciation to form the feature vectors for clustering. One study used translations of sixteen words in each language as characteristic. Their results had big correspondence to genetic language classification [4]. In contrast, for a study that uses typological features of languages as data, the results turned out to be very different from genetic classification [9].

3 Dataset and Experimental Set Up

This section goes into detail about the method of this research. Figure 3 displays an overview of the method.

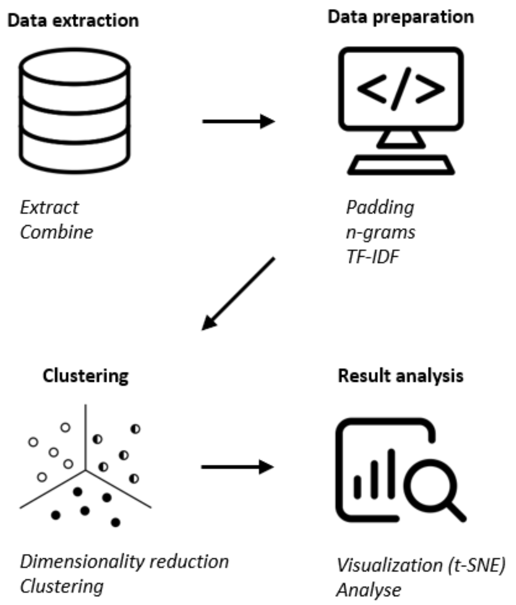


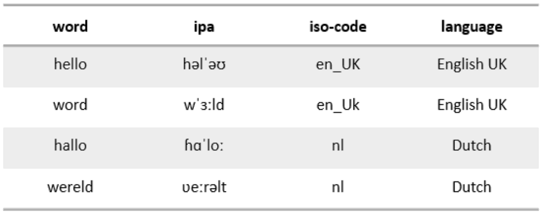
Figure 3: Summary of methodology

3.1 Data extraction

The required data consists of documents of words written in IPA for as many languages as possible. No such ready made dataset exists, hence there was a need to create one. In order to do this, the data from ipa-dict [5] and Wiktextract [27] were used. Ipa-dict contains of lists of word to pronunciation mappings for 24 languages. There are multiple formats available for the dictionaries. The CSV format was opted for, due to easy integration with the Pandas package. Wiktextract is a Python package that provides the functionality of interaction with data from Wiktionary. Wiktionary is an extensive dictionary that contains a lot of information about words from many languages. Among that information, often the pronunciation of words can be found [7]. Wiktextract provides Wiktionary data dumps in JSON format. These were used to acquire the IPA forms of words. Each word with available pronunciation information was added to a CSV file in the format:

word , *pronunciation* , *language code*

For some words, multiple pronunciations are available. In this case, the first option was taken. This, to make sure that each word is equally represented in the data set. As a next step, the two CSV files that are described above were read and combined to one big dataset displayed in a Pandas data frame. By combining languages present in both sources, duplicates arise. This problem was solved by always taking the value provided by ipa-dict in case of a duplicate. Furthermore, the ipa-dict has data for multiple versions of English (US and UK), Spanish (Regular and Mexican) French (Regular and Quebec), Vietnamese (Northern, Southern and Central) and Chinese (Simplified and original). It was not possible to make the same distinction between dialects for the words of these languages extracted from Wiktionary. Therefore, the Wiktionary data was omitted and thus only the ipa-dict data was used for these languages. Lastly, inspection of the pronuncation data showed quite some symbols and signs other than those from the IPA in the pronunciation. As the inconsistencies came from the data for the Persian, Arabic and Japanese language, those languages were removed from the dataset. The final result is a large dataset with the structure displayed in Figure 4. where the combination of word and language form a unique key to the pronunciation.

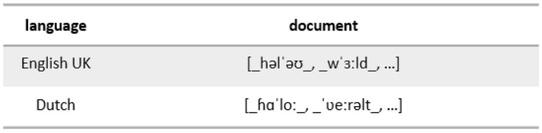


word	ipa	iso-code	language
hello	həlˈəʊ	en_UK	English UK
word	wˈɜːld	en_Uk	English UK
hallo	ɦɑˈloː	nl	Dutch
wereld	ʋeːrəlt	nl	Dutch

Figure 4: Example dataset entries

3.2 Data preparation

After the dataset creation, a few preparation steps follow. The first step is to surround each pronunciation by underscores, to display the start and end. The reason for this is the fact that the order and the place of residence of phonemes in words are very relevant for how a language sounds. As a next step the words were grouped per language and combined into one big data container, in NLP often referred to as a document for clustering (Figure 5).



language	document
English UK	[_həlˈəʊ_, _wˈɜːld_, ...]
Dutch	[_ɦɑˈloː_, _ˈʋeːrəlt_, ...]

Figure 5: Data grouped per language

The amount of data available per language varies from a few up to tens of thousands of entries. To make sure a language is reliably represented a minimum of hundred words was maintained. Languages with less data were not used for clustering, for which 250 languages remained. From these the Indo-European languages were extracted, ending up with 62 languages for which the data quantities are displayed in Figure 6 with a logarithmic scale for the size.

Next, the n-grams are formed. For this research a range of one up to and including four was used. The choice for this range was based on the intuitive thought of an n-gram representing a sound or sound combination which is relatively short in length. This supports the goal of finding patterns in sounds instead of whole words. Accompanied by the fact that it is also a range commonly found in research [1]. Figure 7 gives the result of the first preparation steps.

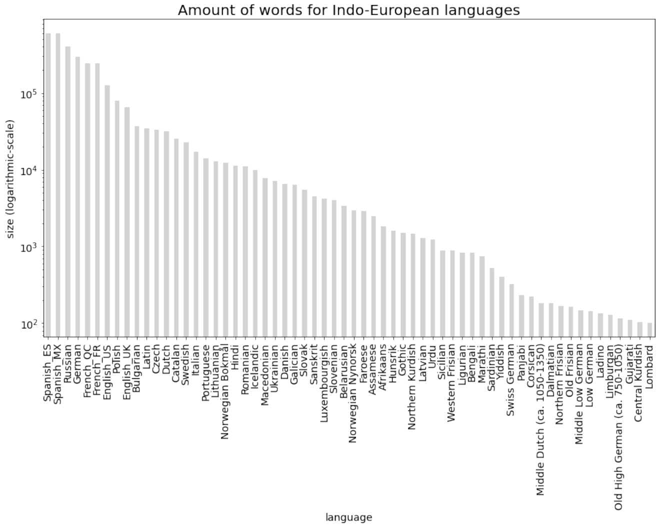
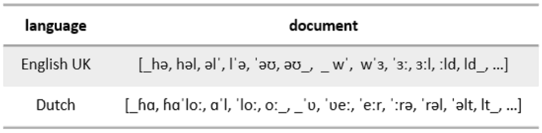


Figure 6: Data size Indo-European languages



language	document
English UK	[_hə, həl, əlˈ, lˈə, ˈəʊ, əʊ_, _wˈ, wˈɜ, ˈɜː, ɜːl, ːld, ld_, ...]
Dutch	[_ɦɑ, ɦɑˈloː, ɑˈl, ˈloː, oː_, _ˈʋ, ˈʋeː, ˈeːr, ˈːrə, ˈrəl, ˈəlt, lt_, ...]

Figure 7: N-grams in documents per language

Finally, the data is structured with TF-IDF using Scikit Learn's tfd-if vectorizer. The calculation of a TF-IDF weight ($w_{t,l}$) for term t and language l consists of two parts: term frequency and inverse document frequency. [21]. For both parts various implementations exists that slightly differ. However, the core principle of representing term importance is not affected. Note that in this case a term is an n-gram. The basic definition of term frequency ($tf_{t,l}$) is the number of occurrences of term t in language l. Considering the unlikeliness that a n-gram occurring ten times more often, has ten times higher importance [20], we apply sub-linear scaling [16]. In this research sub-linear scaling is applied in terms of logarithmic scaling in the following way:

$$\log tf_{t,l} + 1$$

Moving on to the inverse document frequency. The default implementation of Scikit-Learn uses the following formula:

$$idf_{t,l} = \log \frac{n_l}{df_t + 1}$$

Combining the two results gives:

$$w_{t,l} = (\log tf_{t,l} + 1) * \log \frac{n}{df_t + 1}$$

The vectors per language (v_l) formed with all the weights calculated using the described formula, are then normalized by the Euclidean norm:

$$\mathbf{v}_{final} = \frac{\mathbf{v}_l}{\|\mathbf{v}_l\|_2}$$

The normalization is part of the standard Scikit-Learn implementation as well and accounts for data imbalance per language. After this process of TF-IDF, the resulting matrix is structured like Figure 8. There are 62 rows, one for each language and one column for each unique n-gram that occurs over all documents.

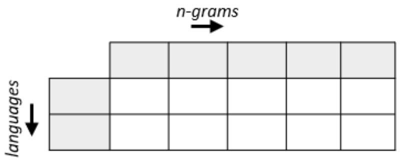
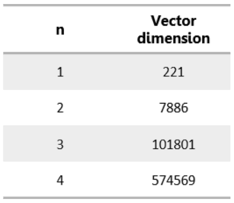


Figure 8: TF-IDF matrix

3.3 Dimensionality Reduction

For this research, the data points used for clustering are the rows of the TF-IDF matrix. So there is a vector per language with TF-IDF weights for each n-gram as values. As a first step of the clustering process, dimensionality reduction is applied. Since the IPA uses a lot of symbols to denote the pronunciation the amount of optional combinations for n-grams rises quickly. As a consequence, the dimensions of the vectors become very high (see Figure 9). Using SVD these dimensions are reduced to 50. This eases discovery of relationships between the n-grams that are analyzed. It allows the algorithms to consider fewer random variables and establish clearer links between those still present.



n	Vector dimension
1	221
2	7886
3	101801
4	574569

Figure 9: TF-IDF vector dimensions for varying n in n-grams

3.4 K-means

The classification of the Indo-European languages from linguistics based on origin defines four big groups of languages: Germanic, Italic, Balto Slavic, and Indo-Iranian. See appendix A for the languages used and the division of those languages over the four language groups. In order to see if there is a clear and easy to find overlap with the linguistics classification, k-means clustering with $k = 4$ was applied as a baseline.

The classification of k-means clustering and language classification, is actually a partition of the set of Indo-European language. The two partitions are compared to investigate the connection between them using the Adjusted Rand Index (ARI). Also, the silhouette score is calculated. The motivation for using the latter metric will become clear in the result section. As was mentioned in Chapter 2, the initial step of k-means is random initialisation of k cluster centres. To account for the randomness the set of actions of running the k-means algorithm and the calculation of silhouette score and the ARI was repeated 100 times. This in order to find minimum, maximum and average values for the metrics.

Furthermore, a sanity check was done by adding languages very different from the Indo-European to see how those languages compare to the Indo-European languages. Note that adding the languages to the data set means that the procedure of vector creation using TF-IDF and dimension reduction has to be gone through as well. This is due to the fact that adding new languages influences the inverse document frequency part of TF-IDF. Inverse language frequency might function as a more explanatory name in this case.

Lastly, the elbow method was applied to see which k is optimal for k-means clustering of the pronunciation data and how that corresponds to linguistics.

3.5 Distance metric

The thing that all clustering algorithms have in common is the use of a distance metric. Many options exist, but for text clustering Euclidean and Cosine distance are most often used [10]. Euclidean distance measures the length of the line between two data points. The Cosine distance measures the angle between two vectors [22]. With normalized vectors, the square of the euclidean distance is equal to the cosine distance [8]. Since squaring is a monotonic transformation for positive numbers it will not change the result when trying to find the nearest cluster centre for each point in k-means. Therefore, the use of either Cosine or Euclidean distance does not influence the final clusters. We opted for Euclidean distance, since this is standard for the k-means clustering in the Scikit-Learn package.

4 Results and discussion

As outlined in the previous chapter, the dataset selected for this study comprises pronunciation data of 250 languages from which 62 are Indo-European languages. The latter are converted into a TF-IDF matrix representation and used for clustering. This section provides findings and results obtained from the analysis of the conducted experiments which are immediately discussed. A more general discussion will follow in the next Chapter. In this section there will be attention for the linguistic classification first. Then, the outcome of k-means clustering will be shown and analyzed followed by the evaluation of non-Indo-European languages to the dataset.

4.1 Linguistic classification

In the t-SNE plots of Figure 10, each language is a data point and each colour represents a language group as specified by linguistic classification based on origin. In each of the four plots it is promising to see that the colours show some clustering behaviour as opposed to being spread over the whole surface. So, it seems possible for a clustering algorithm to pick up clusters. Furthermore, based on the visualisation, it looks like the data is quite uniformly spread over the feature space. The presence of distinct clusters is missing, which might complicate picking up the clusters for the clustering algorithm.

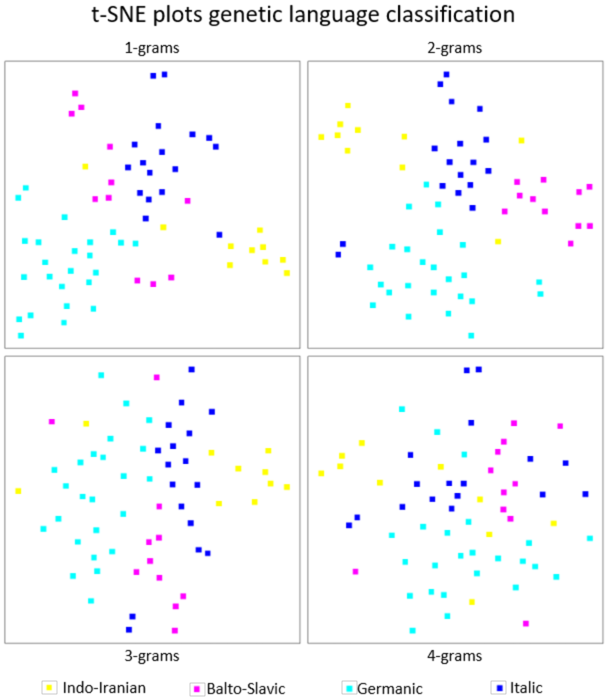
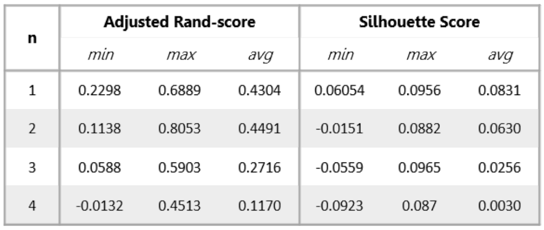


Figure 10: Linguistic clustering

4.2 K-means

Applying the k-means with $k = 4$ results in four clusters that are compared to the linguistic classification using the ARI. As can be seen in figure 11, the range for the adjusted rand score per type of n-gram is relatively big considering the full range of -1 to 1. The wide range indicates that between each iteration, there is large variation in the outcome of the clustering. As a consequence the correspondence with the linguistic classification changes each iteration as well.



n	Adjusted Rand-score			Silhouette Score		
	min	*max*	*avg*	*min*	*max*	*avg*
1	0.2298	0.6889	0.4304	0.06054	0.0956	0.0831
2	0.1138	0.8053	0.4491	-0.0151	0.0882	0.0630
3	0.0588	0.5903	0.2716	-0.0559	0.0965	0.0256
4	-0.0132	0.4513	0.1170	-0.0923	0.087	0.0030

Figure 11: Metrics obtained from 100 iterations of k-means clustering

Silhouette score

As already mentioned, the t-SNE plots suggest a relatively even spread of the data over the feature space as opposed to the formation of clear and coherent clusters. Since t-SNE makes assumptions about the data that differ from the assumptions clustering algorithm make, one should be careful to jump to conclusions too quickly only based on the visualisation. However, the absence of clear clusters is also verified when looking at the silhouette score displayed in figure 11. The silhouette score values around zero indicate cluster indifference, which is what is expected with even spread of the data. The combination of the randomness in the initialisation of the k centroids in k-means and the relative equal distribution of data within the feature space increases the importance of the initially picked centroids on the result of the clustering. This would be a logical explanation for the variation in classes over the various iterations of applying k-means.

Adjusted Rand Index

Nonetheless, the similarity of the partitions based on the ARI indicate similarity in classification of linguistics and k-means. All the variations of n-grams have an ARI above zero, which would be the expected value when comparing two random partitions. The partition of the languages obtained using clustering with TF-IDF vectors based on 1-grams and 2-grams have the biggest similarity with the partitions that linguistics specify. One explanation for this is that 1-grams and 2-grams are the most suitable for the representation of pronunciation when comparing it to genetic origin. However, it could also be that the dimension of 50 for 3-grams and 4-grams are too low for a clustering algorithm to pick up patterns. The relative reduction in dimensionality of the TF-IDF vectors for 3-grams and 4-grams is much bigger compared to those of 1-grams and 2-grams.

ARI interpretation

In Figure 12 and 13 the similarity of the linguistic classification and the classification of the k-means clustering is visualized for two different values of the ARI. This helps interpreting the value of the ARI and allows for more detailed analysation. The colours of the squares correspond to the linguistic classification. The position and colours of these squares is exactly the same as in the top right plot in Figure 10. The little circles situated on top of the squares are for the classification found by k-means. The colours of the circles are assigned in such a way that the amount of data points with the same colour for the square and for the circle is maximized. That is, each cluster obtained from k-means is assigned the colour of the linguistic language group with which it has the biggest overlap of languages.

The ARI for Figure 12 is 0.8053, the maximum value reached in all the iterations of k-means as can be seen in Figure 11. Here, five languages are classified differently by k-means as compared to linguistics. Two pairs are strongly related since they are two dialects of both the French and Kurdish language. It is desirable that two dialects cluster together considering the aim of finding language similarity. However, the question remains why languages classify differently when compared to linguistic classification. By evaluation of the other cluster results, it appears that some lan-

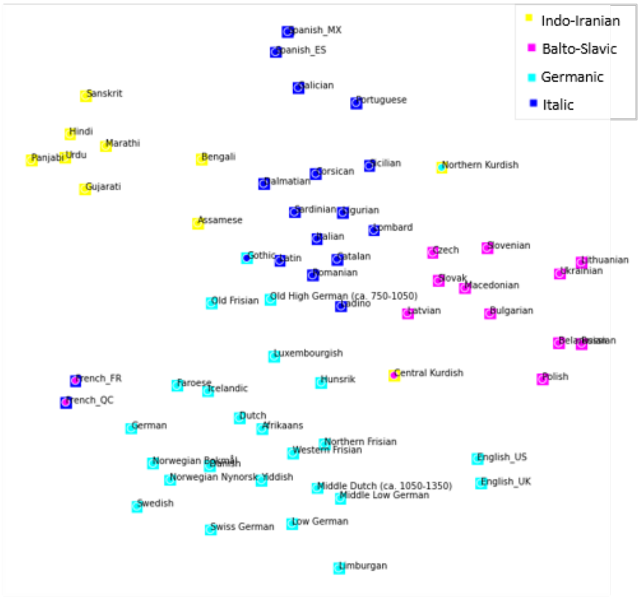


Figure 12: Similarity of k-means and linguistic classification with ARI of 0.8053

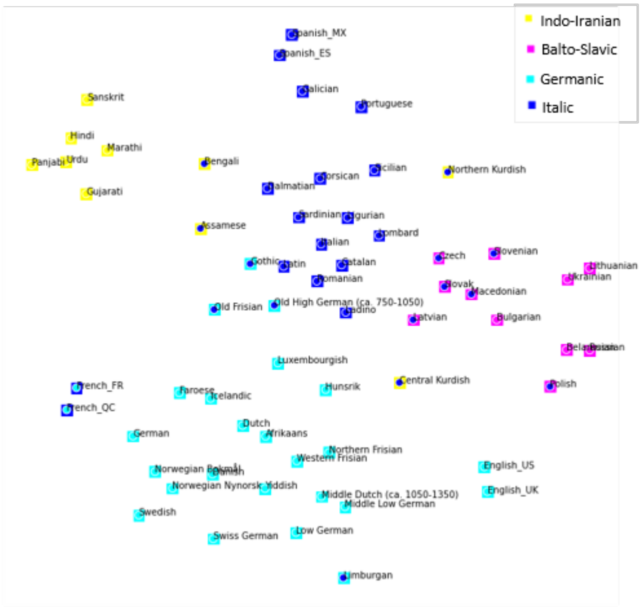


Figure 13: Similarity of k-means and linguistic classification with ARI of 0.4139

guages are often classified differently compared to linguistic classification. The five differently classified languages from Figure 12 belong to that category. In the plot the datapoints corresponding to those five languages are on the edges of the formed clusters. So, most probably the way they are classified is very sensitive to the initialisation of the cluster centroids. This reasoning looks solely at the actual clustering. However, the translation of being on the edge of a cluster to linguistics, is that the pronunciation of those languages is on the border between various language families. This could have many causes, one of which is geographical location. This possibly applies for French for example, since France is close to many countries that do not have languages from the Italic language family. However, this is one explanation and many more options exist.

The value of the rand score for Figure 13 is 0.4138, which is around the average ARI for 1-grams and 2-grams. In this plot we see that sixteen languages are classified differently, which is approximately a quarter of the size of the total dataset. In this specific example many Balto-Slavic languages are clustered together with a large group of Italic languages. However, in other iteration of the k-means the variations are in other language groups, so no clear patterns on how languages classify differently between the various language groups have been found. Within the clusters of k-means patterns do arise. There are groups of languages that are almost always classified together by k-means, indicating that there is a strong connection between their sounds. This will be further mentioned in the future work section.

4.3 Addition of non-Indo-European languages

So far, it seems like similar languages based on origin are more likely to cluster together. The expected behaviour for languages very different from the Indo-European languages, is to be distant. A sanity check is done by adding, Korean, Finnish and two dialects of Chinese. The plots in Figure 14 show that the distance is there, especially for Chinese. The other two languages blend more into the Indo-European languages. From the t-SNE plots Finnish is most similar to Indo-European languages, which we expected since it is a European country surrounded by countries that speak Indo-European languages.

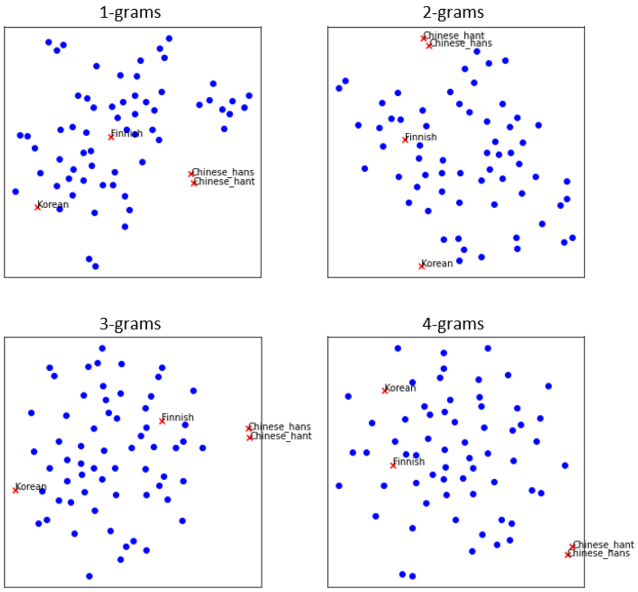


Figure 14: t-SNE plots including four not Indo-European Languages

4.4 Elbow method

Lastly, the lines in Figure 15 do not have an inflection point. Therefore the elbow methods is inconclusive about which k optimal for the k-means clustering. When there is large agreement between linguistic classification and k-means, a value of four would be expected. Or a higher value, which would mean that smaller subgroups are recognized. However, the inconclusive can be led bag to the even data spread and is therefore not surprising. It does indicate that the choice for the value of k has to be based on other arguments. Like for this research we initially chose four based on genetic classification. But if one wants to do research on more specific subgroups within Indo-European languages another value could be suitable.

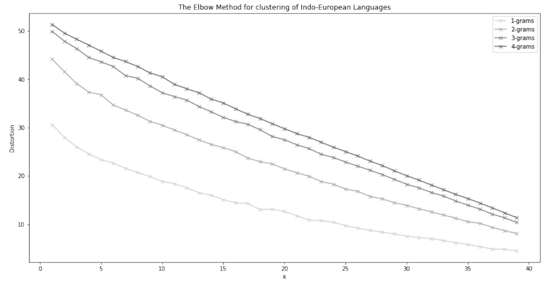


Figure 15: Elbow method for all variations of n-grams

5 Discussion

This section discussed the conducted experiments and the obtained results with a global look on the research.

From the experiments, it becomes clear that a relation can be found between the linguistic and k-means language classification. To which extent this relation applies is harder to answer. It varies between the iterations of k-means and with different tuning options like the n in n-grams. The absence of an objective measure for similar sounding languages complicates the argumentation for why and to what extent languages are classified differently. Moreover, the validity of the results will be improved by addition of data. Although the vectors normalisation accounts for the variation in the data size per language, the languages with few data available will still benefit due to better representativeness. Also, a specific niche was looked at using Indo-European languages and k-means clustering. The findings based on this approach do not necessarily extent to language families.

Lastly, the initial idea was to also conduct a hierarchical approach since within the four big language groups there exist more specific subgroups. A hierarchical approach is suitable to try since it decomposes the dataset in a hierarchical structure [19]. The tree structure defined by linguistics can be found in appendix B. We tried the hierarchical approach. One of the results (see Appendix C) came out very similar to the linguistic classification, even on a more specific level. However, the overall results of hierarchical clustering were only analysed manually. We considered the results were not of enough value to be described in this paper, therefore it is left for future research.

6 Future Work

This research gives rise to many opportunities for future research. Either by doing a more in depth analysis of the approach used in this research or by trying something completely different. More in depth analysis should focus on the variations in the iterations of k-means. Finding groups of languages that occur together most of the time and finding languages that occur are on the edges between language groups. We speculate that there is a specific distance between two vectors for which there is a very high probability of being of the same origin. For a different approach we recommend a hierarchical clustering algorithm. As already mentioned in the previous section the foundations for this can be taken from this research. Furthermore, extension of the dataset especially for languages with not to much data available will result in more representative vectors. Moreover, other language families could be considered to see how they relate to genetic classification. Lastly, research can be done to define the applicability and usefulness of the found classification method.

7 Conclusion

In this research, we composed a dataset with pronunciation of word written in IPA for 250 languages. Using this data, we compared data driven language classification using the IPA and clustering to linguistic language classification. The results show that there are definitely patterns in pronunciation that give rise to similarity in k-means and linguistic language classification. Due to the relative even spread of the data over the feature space, the variation in obtained results between various iterations of the k-means clustering was large and no optimal value for k could be found. However, for 1-grams and 2-grams the average Adjusted Rand Index was 0.4178 and 0.4363 respectively, which translates to approximately three quarters of the languages classified in groups that are in agreement with linguistics. Still, this research can be much more extended and functions as a foundation and a source of inspiration for future research.

8 Responsible Research

This research was done as part of the course CSE3000 at Delft University of Technology. To accommodate full reproducibility every step taken has been described as precise as possible. All the settings and parameters are enclosed and the code is openly available on the Github Repository[1]. The data has been extracted from sources with open licence and was handled with care. No data has been purposely removed unless a justified explanation was provided. This explanation is always related to better representativeness and not to manipulate results. The data has only been handled objectively and no outliers were removed. Probability did play a role in the experimental setup. Therefore, some result might come out slightly different when the experiment is repeated. However, the results should not differ significantly. The general conclusions drawn from the results should therefore not be any different when the research is repeated.

[1] https://github.com/i-rethans/CSE3000_language_similarity

References

[1] Abdelmalek Amine, Zakaria Elberrichi, Michel Simonet, and Mimoun Malki. Wordnet-based and n-grams-based document clustering: A comparative study. pages 394 – 401, 12 2008.

[2] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, 59:44–58, 2020.

[3] Shaeela Ayesha, Muhammad Kashif Hanif, and Ramzan Talib. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, 59:44–58, 2020.

[4] Vladimir Batagelj, Tomaz Pisanski, and Damijana Keržič. Automatic clustering of languages. *Computational Linguistics*, 18:339–352, 01 1992.

[5] Liam Doherty. ipa-dict. https://github.com/open-dict-data/ipa-dict, 10 2019.

[6] The editors of Encyclopæ dia. International phonetic alphabet, 05 2020.

[7] Wikimedia Foundation. Wiktionary, Jan 2021.

[8] Stephen L. France, J. Douglas Carroll, and Hui Xiong. Distance metrics for high dimensional nearest neighborhood recovery: Compression and normalization. *Information Sciences*, 184(1):92–110, 2012.

[9] Ryan Georgi, Fei Xia, and William Lewis. Comparing language similarity across genetic and typologically-based groupings. volume 2, pages 385–393, 01 2010.

[10] J. Ghosh and A. Strehl. *Similarity-Based Text Clustering: A Comparative Study*, pages 73–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[11] Jirka Hana, Anna Feldman, and Chris Brew. A resource-light approach to russian morphology: Tagging russian using czech resources. pages 222–229, 01 2004.

[12] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[13] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, page 604–613, New York, NY, USA, 1998. Association for Computing Machinery.

[14] Alan Jeffares. K-means: A complete introduction, Nov 2019.

[15] Ammar Ismael Kadhim, Yu-N Cheah, and Nurul Hashimah Ahamed. Text document preprocessing and dimension reduction techniques for text document clustering. In *2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology*, pages 69–73, 2014.

[16] Yoon Kim and Owen Zhang. Credibility adjusted term frequency: A supervised term weighting scheme for sentiment analysis and text classification, 2014.

[17] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003. Biometrics.

[18] John P. Lyons, Pavle P. Ivić, and Eric P. Hamp. Linguistics, 09 2020.

[19] T. Soni Madhulatha. An overview on clustering methods. *CoRR*, abs/1205.1117, 2012.

[20] G. Paltoglou and M. Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *ACL*, 2010.

[21] Shahzad Qaiser and R. Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181:25–29, 2018.

[22] Gang Qian, Shamik Sural, Yuelong Gu, and Sakti Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, SAC '04, page 1232–1237, New York, NY, USA, 2004. Association for Computing Machinery.

[23] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[24] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[25] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010.

[26] Fei Xia and William Lewis. Multilingual structural projection across interlinear text. pages 452–459, 01 2007.

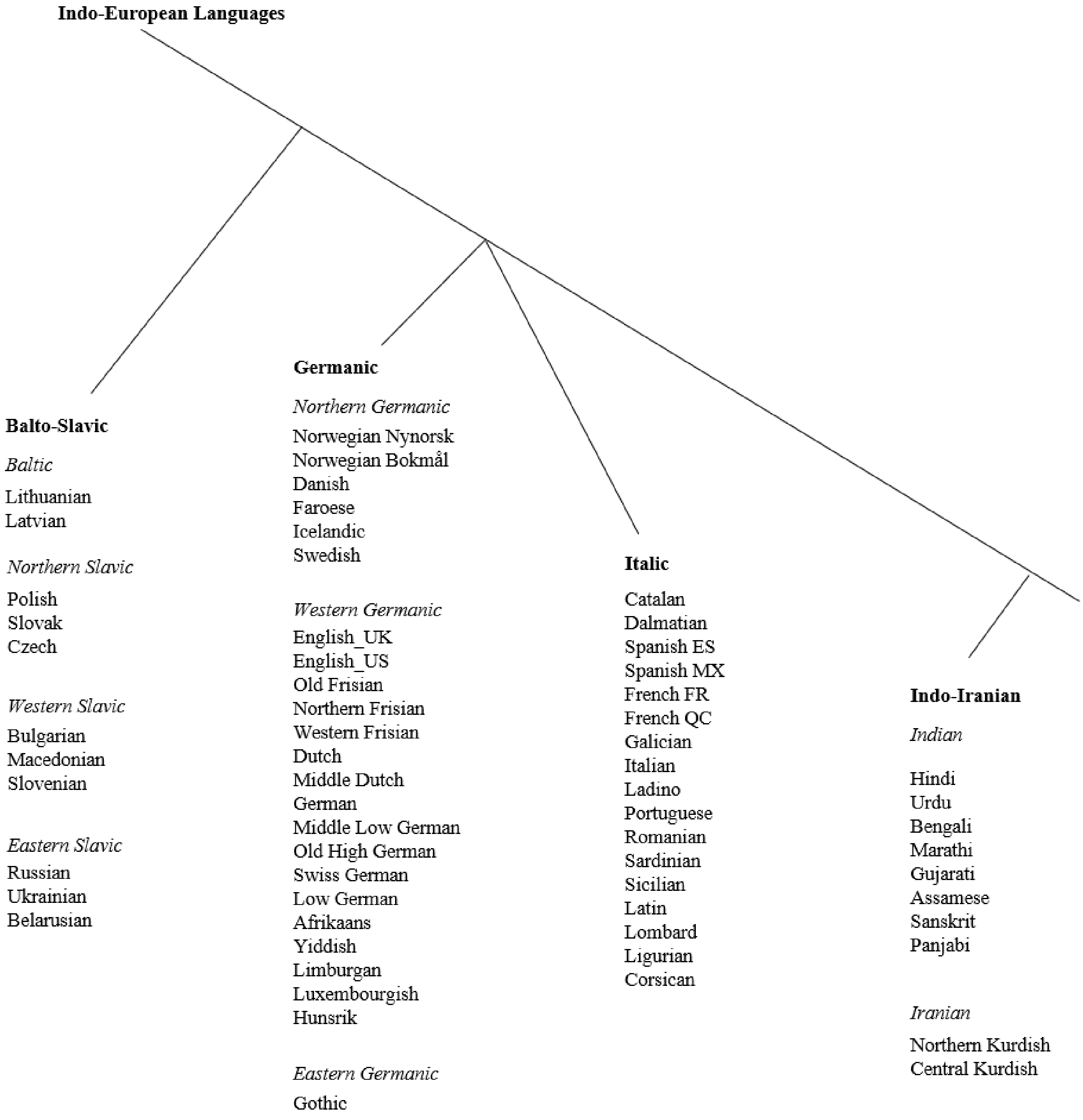
[27] Tatu Ylonen. wiktextract. https://github.com/tatuylonen/wiktextract, 04 2021.

[28] Chunhui Yuan and Haitao Yang. Research on k-value selection method of k-means clustering algorithm. *J*, 2(2):226–235, 2019.

A Indo-European Language Groups

Balto-Slavic	**Germanic**	**Italic**	**Indo-Iranian**
Russian	Hunsrik	Lombard	Marathi
Belarusian	German	Ligurian	Gujarati
Polish	Luxembourgish	Dalmatian	Sanskrit
Ukrainian	Swiss German	Ladino	Bengali
Lithuanian	Dutch	Sicilian	Assamese
Macedonian	Afrikaans	Sardinian	Urdu
Bulgarian	Western Frisian	Corsican	Hindi
Slovak	Low German	Portuguese	Panjabi
Czech	Northern Frisian	Galician	Northern Kurdish
Slovenian	Yiddish	Romanian	Central Kurdish
Latvian	Middle Low German	Italia	
	Middle Dutch	Latin	
	Limburgan	Catalan	
	Old Frisian	Spanish MX	
	Gothic	French QC	
	Old High German	French FR	
	Norwegian Nynorsk		
	Norwegian Bokmål		
	Swedish		
	Icelandic		
	Faroese		
	English US		
	English UK		
	Danish		

B Language tree of the Indo-European language family



Indo-European Languages

Balto-Slavic

Baltic

Lithuanian
Latvian

Northern Slavic

Polish
Slovak
Czech

Western Slavic

Bulgarian
Macedonian
Slovenian

Eastern Slavic

Russian
Ukrainian
Belarusian

Germanic

Northern Germanic

Norwegian Nynorsk
Norwegian Bokmål
Danish
Faroese
Icelandic
Swedish

Western Germanic

English_UK
English_US
Old Frisian
Northern Frisian
Western Frisian
Dutch
Middle Dutch
German
Middle Low German
Old High German
Swiss German
Low German
Afrikaans
Yiddish
Limburgan
Luxembourgish
Hunsrik

Eastern Germanic

Gothic

Italic

Catalan
Dalmatian
Spanish ES
Spanish MX
French FR
French QC
Galician
Italian
Ladino
Portuguese
Romanian
Sardinian
Sicilian
Latin
Lombard
Ligurian
Corsican

Indo-Iranian

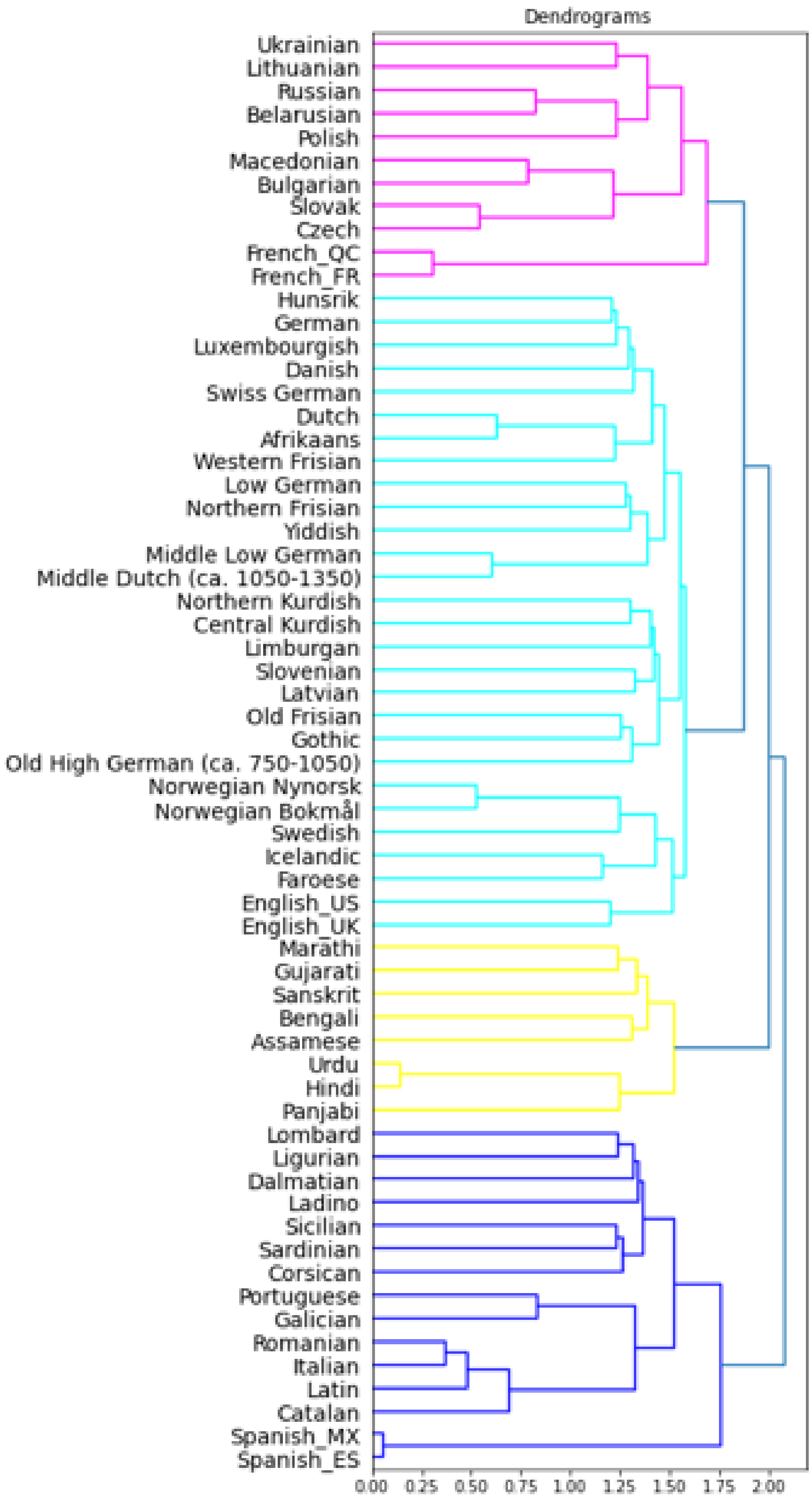
Indian

Hindi
Urdu
Bengali
Marathi
Gujarati
Assamese
Sanskrit
Panjabi

Iranian

Northern Kurdish
Central Kurdish
]

C Hierarchical clustering result


Dendrograms
Ukrainian
Lithuanian
Russian
Belarusian
Polish
Macedonian
Bulgarian
Slovak
Czech
French_QC
French_FR
Hunsrik
German
Luxembourgish
Danish
Swiss German
Dutch
Afrikaans
Western Frisian
Low German
Northern Frisian
Yiddish
Middle Low German
Middle Dutch (ca. 1050-1350)
Northern Kurdish
Central Kurdish
Limburgan
Slovenian
Latvian
Old Frisian
Gothic
Old High German (ca. 750-1050)
Norwegian Nynorsk
Norwegian Bokmål
Swedish
Icelandic
Faroese
English_US
English_UK
Marathi
Gujarati
Sanskrit
Bengali
Assamese
Urdu
Hindi
Panjabi
Lombard
Ligurian
Dalmatian
Ladino
Sicilian
Sardinian
Corsican
Portuguese
Galician
Romanian
Italian
Latin
Catalan
Spanish_MX
Spanish_ES