8-2022

Feature Analysis of Indus Valley and Dravidian Language Scripts with Similarity Matrices

Sarat Sasank Barla

Sai Surya Sanjay Alamuru

Peter Revesz

Feature Analysis of Indus Valley and Dravidian Language Scripts with Similarity Matrices

Sarat Sasank Barla, Sai Surya Sanjay Alamuru,
& Peter Z. Revesz

School of Computing, University of Nebraska–Lincoln

Contacts — sbarla2@huskers.unl.edu ; salamuru2@huskers.unl.edu ; revesz@cse.unl.edu

Barla, Alamuru, & Revesz in 26th IDEAS 2022

Abstract

This paper investigates the similarity between the Indus Valley script and the Kannada, Malayalam, Tamil, and Telugu scripts that are used to write Dravidian languages. The closeness of these scripts is determined by applying a feature analysis of each sign of these scripts and creating similarity matrices that describe the similarity of any pair of signs from two different scripts. The feature list that we use for the analysis of these Dravidian language-related scripts includes six new features beyond the thirteen features that were used for the study of Minoan Linear A and related scripts by Revesz. These new features are the check mark, short vertical line, dot, upper curve, parallel curves, and horizontal line features.

CCS Concepts: Information systems→Information systems applications; Data mining.

Keywords: Dravidian, Epigraphy, Feature analysis, Indus Valley script, Script similarity measure, Sumerian pictogram

Published in *International Database Engineered Applications Symposium (IDEAS'22), August 22–24, 2022, Budapest, Hungary*. ACM, New York, NY, USA.
doi:10.1145/3548785.3548801

1 Introduction

It is strongly believed by most of the people that the first human civilization flourished somewhere near the present day upper eastern part of Africa and that all humanity at that time used to speak a single language called a protolanguage, which is the origin of all the languages spoken in today's world [13]. The protolanguage spread and diversified together with human populations as humans started to leave the Sahara when the temperatures started soaring and the desertification of the Sahara begun. The desertification prompted people to split into small groups and to travel to different places in search of food, shelter, and viable climatic conditions. This process resulted in a change in the living style of people along with their environmental needs, requirements, and way of communicating. Although many scientists and researchers believe in the concept of divergence of languages from a protolanguage, this hypothesis is still controversial. Finding how similar two languages is a complex problem. The following are three major ways which help us determine how closely languages are related.

1.1 Human migrations

In this method, we try tracking people's migration throughout history and observe how does this migration affected the languages. Generally, the scientists relate linguistics to molecular biology. From the concept of tracking the mitochondria present inside the nucleus of the human body one can trace back people's ancestors, and research suggests this process also works well for finding the language path. However, we cannot completely rely on our process in this method, since when starting to go far back in time we will have less evidence and no accurate metrics on which to base our assumptions.

1.2 Similar sounding words

We know that there are many languages that are derived from others which contain the same words which convey similar meanings. However, there is a very high probability of a word with the same sound having a different meaning. These are known as homophones. For example, the word *filter* in English coveys a meaning of a substance which is used to separate different things, but the same word means '*poison*' in French. Such words are false cognates. Hence simply looking for similar sounding words is a faulty method.

1.3 Feature analysis

In this approach, we find the similarity between two languages by observing the similarity between the scripts and their regular changes. This process is done by developing features which represent all the letters in the scripts and developing the feature evaluation table. When we have the feature analysis tables for at least two languages we can create the similarity matrix to check how close the two scripts are related. We follow this method in our implementation process.

2 Background

The Indus Valley Script is an ancient script developed by the Indus Valley civilization, which existed c. 3500–1900 BCE. The Indus Valley Civilization was first identified at Harappa and Mohenjo-Daro in 1921 and 1922, respectively [7]. The first publication of the seal with Harappan symbols were produced in 1875 in the drawings of Sir Alexander Cunningham. Mahadevan [5] proposed a list of signs with 417 distinct symbols in 1977. Later, the Corpus of Indus Seals and Inscriptions (CISI) introduced 386 different symbols [4, 6, 7].

The Indus Valley Civilization originated during the same period as the Sumerian civilization. The Indus Valley and its river tributaries provided basic food and transportation to the people like the Euphrates and the Tigris Rivers in Mesopotamia. The Indus Valley civilization had brick homes, baths, and forts, and used copper and bronze metals to make tools and weaponry. Different seals were used for commerce which were attached to trade goods and showed a mix of symbols. The most important settlement areas were Mohenjo-Daro and Harappa which contained about 35,000 people. Much research showed evidence of trade between Indus Valley and Mesopotamia [12].

The Dravidian language family represents about thirty languages that are common today in Southern India, including the Kannada, Malayalam, Tamil, and Telugu [14]. Daggumati and Revesz [1–3] suggests the possibility of the migration of proto-Dravidian people to the Indus Valley from Mesopotamia because Sumerian pictograms are the most like Indus Valley Script signs among a set of ancient scripts. In addition, Proto-Dravidian *piru* and Mesopotamian *pirus* both mean ‘elephant’ [12]. The prevalence of Dravidian cognates in the Rig-Veda suggests that Dravidian and Aryan speakers had merged into one language in the large Indo-Gangetic Plain by the time of its composition, while independent Dravidian groups had moved to the boundary of the Indo-Aryan area. The history of Dravidian language evolution is hard to study because the earliest Tamil inscriptions, which were found in the Madurai and Tirunelveli districts of Tamil Nadu, date only from the 2nd century BCE. Perhaps the decipherment of the Indus Valley script could shed more light on the evolution of Dravidian languages.

3 A New Feature Analysis Method

In this paper, we follow the third method of finding similarity among scripts, that is, by using feature analysis and similarity matrices.

3.1 Feature Analysis

The concept of developing features and thereby presenting the results using similarity matrices is initially suggested by Revesz [8, 9].

Revesz [9] found thirteen features that seem to commonly occur in various scripts. These thirteen features can distinguish all the signs in various ancient scripts. For example, **Figure 1** shows a feature analysis of the Minoan Linear A script, where features have a symbol (contains curved line: (, contains an enclosed region: O, has a slanted straight line: etc.). Features that are present are marked as red and features that are absent are marked as black. Given feature tables for two different scripts, a similarity matrix can be generated from them, such as for the Linear A script and the Carian alphabet [2]. In a general view, a similarity matrix helps us to visualize how close the two scripts are at a higher level. This similarity matrix is created by calculating the absolute difference between features of a particular letter in one evaluation table to all the features of a letter in the other evaluation table. This process is to be done for all features of each letter in the first evaluation table. The output of

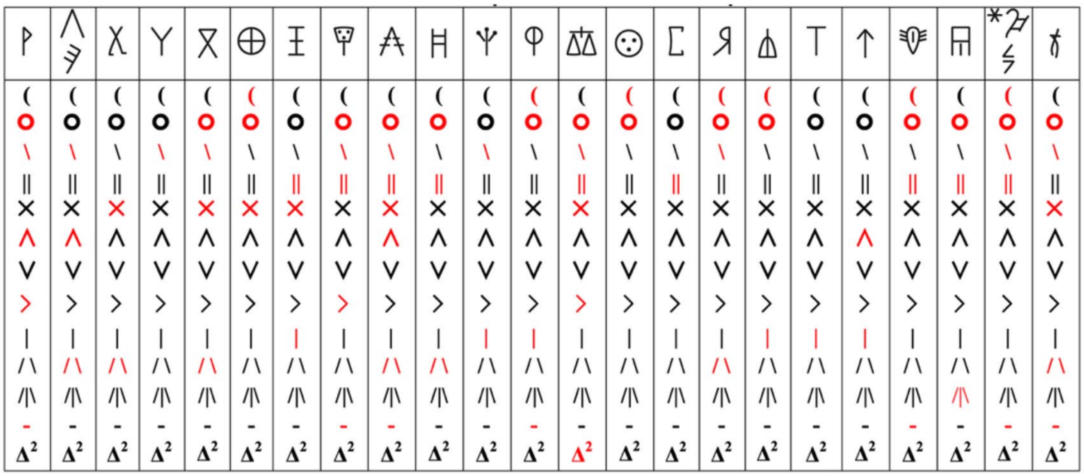


Figure 1 Feature analysis of Linear A signs according to Revesz [9].

this process will be a distance matrix. Then we need to subtract every element in the distance matrix with total number of features, thirteen in this case, to get the similarity matrix.

3.2 Our approach

We have considered the Indus Valley Script and those scripts that are used to write the Dravidian languages of Kannada, Malayalam, Tamil, and Telugu. We applied feature analysis on these languages and try to find similarities among them. We considered 25 of the most common letters from each language and started our process. Unlike western language scripts the Dravidian scripts are more cursive, and we were required to add some extra features to the thirteen features that were proposed in [9]. The new features help to analyze some details of the cursive Dravidian scripts to improve the accuracy of defining the script signs and comparing them. **Figure 2** shows the additional features that we introduced for the sake of an improved analysis.

In Figure 2, the check mark has been a predominant feature in the Telugu scripts and has played a major role in changing the pronunciation of the script signs. In the Kannada, Tamil, and Telugu scripts the presence of a short vertical line, dot, and upper curve have a very different meaning were compared to their absence in the signs of these scripts. The horizontal line in the Malayalam script alone distinguishes

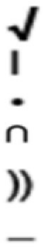


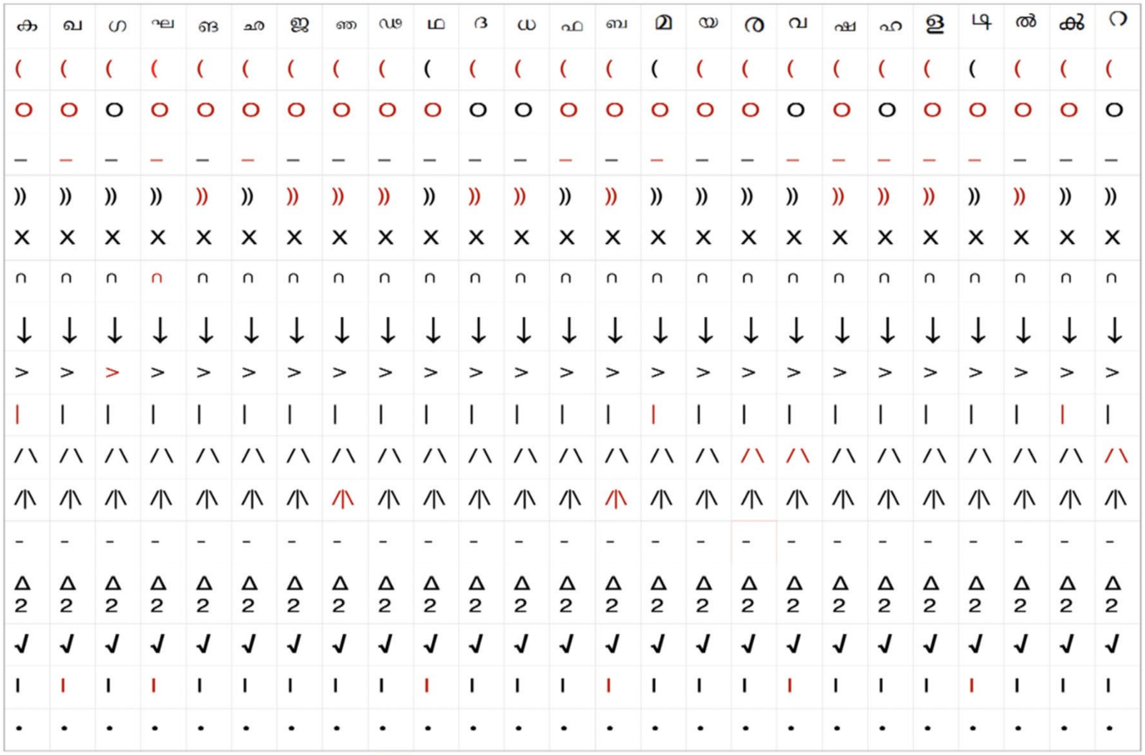
Figure 2 We introduce the following new features from top to bottom: check mark, short vertical line, dot, upper curve, parallel curves, and horizontal line.

more than two signs. Finally, we included parallel curves as these Dravidian scripts are more cursive than the straight-line strokes. For example, there are some Telugu script signs that are differentiated with a single dot mark alone.

After developing these feature analysis tables, we needed to create similarity matrices between any two considered language scripts. This Similarity matrix will be a N x N matrix where N is the number of considered letters for the analysis. Hence, each similarity matrix in our context will be 25 x 25 matrix and contain 625 entries. Therefore, calculating all these entries manually is a very time-consuming process besides being prone to mistakes. Hence, we decided to develop a computer program such that it calculates all the values accurately and effectively. Below we present the process of how we treated the values in the feature evaluation table and used them as inputs in the similarity matrix, together with how we developed the logic for the matrix calculation.

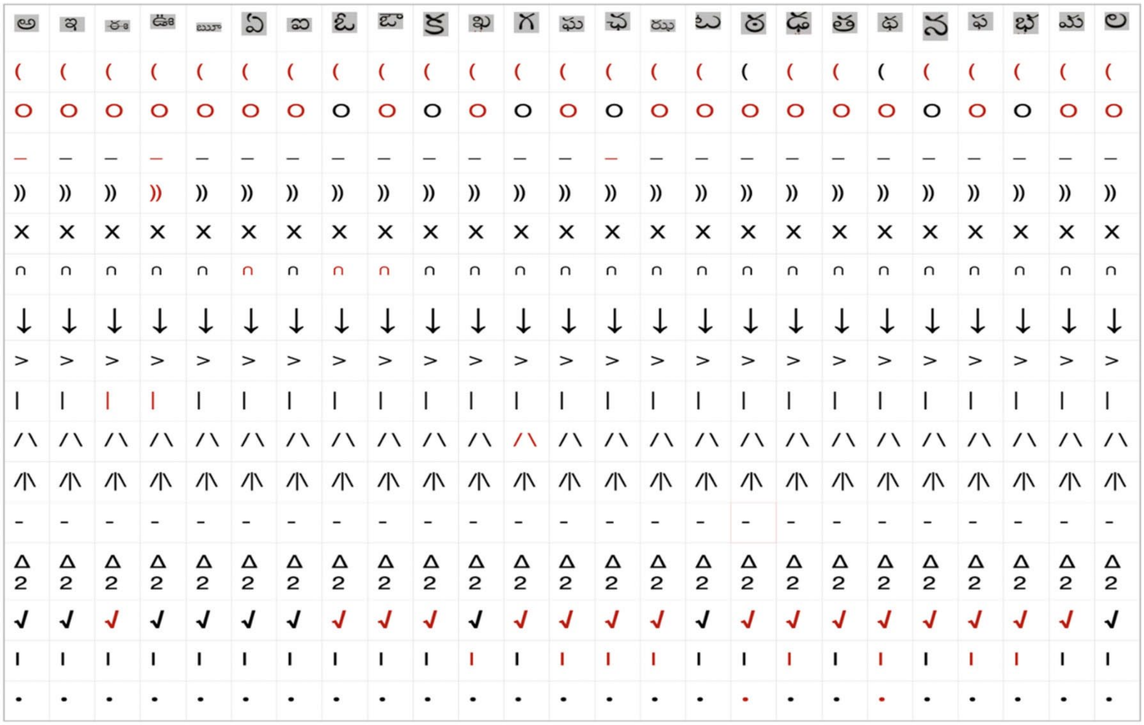
Initially we wanted to consider all the features for a particular sign as a single vector. Hence, the features that are marked red (the features which are present in the letter) are considered as 1's and the remaining black marked features (the features which are not present in that letter) are considered as 0's. Therefore, we can extract a total of 25 vectors (from the 25 signs) from one feature evaluation table. These 25 vectors were compared separately with all other 25 feature vectors of the second feature evaluation table. **Figure 3** shows the feature analysis matrix for the Malayalam script. **Figure 4** shows the feature analysis matrix for the Telugu script.

After the formation of the two feature matrices, we need to transpose one of the matrices to facilitate certain matrix operations. Here we have



ക	ഖ	ഗ	ഘ	ങ	ഛ	ജ	ഞ	ഢ	ഥ	ദ	ധ	ഫ	ബ	മ	യ	ര	വ	ഷ	ഹ	ള	ഺ	ൽ	ൾ	ി
(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(
O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))
×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>
|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|
/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\
/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2
√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Figure 3 Feature analysis of the Malayalam script.



అ	ఇ	ఈ	ఊ	ౠ	ఏ	ఐ	ఓ	ఔ	క	ఖ	గ	ఘ	ఛ	ఝ	ట	ఠ	ఢ	త	థ	న	ఫ	భ	మ	ల
(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(	(
O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))	))
×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩	∩
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>	>
|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|	|
/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\	/\
/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\	/|\
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2	Δ 2
√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı	ı
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Figure 4 Feature analysis of the Telugu script.

two 25 x 16 matrices and since we need to perform multiplication functions during the process of forming a similarity matrix, we will encounter a dimensional mismatch error if we do not transpose one of the two feature vector matrices.

We had everything set to apply our main operation to create the similarity matrix, but the question is what this main operation exactly should be. Before discussing that, let's comprehend and analyze how we form a similarity matrix in the traditional way. We calculate the absolute difference between two features in their respective position and remove this difference from the total features value to get the similarity number. For doing this we initially tried with three methods. One is by using the dot product. We all know that the dot product tells us about the angle between the two vectors ($A \cdot B = A^*B^*\cos(\theta)$) where θ is the angle which determines by how much these two vectors got deviated from one another. When we try implementing this model unlike the real dot product the machine was performing a simple matrix multiplication (a weighted sum of vectors) due to which we tend to lose some of the feature values.

In the second method we try implementing XOR operation on the feature vectors which return value 1 only when there are different corresponding vectors (0 and 1, 1 and 0) which exactly what we expect the result to be. But again, we encountered trouble during its implementation. Applying the XOR operation upon the vectors gives the bitwise XOR results rather than the element-wise results. Due to this, the final matrix has a dimension of 25 x 16 unlike the square matrix 25 x 25 that we expect.

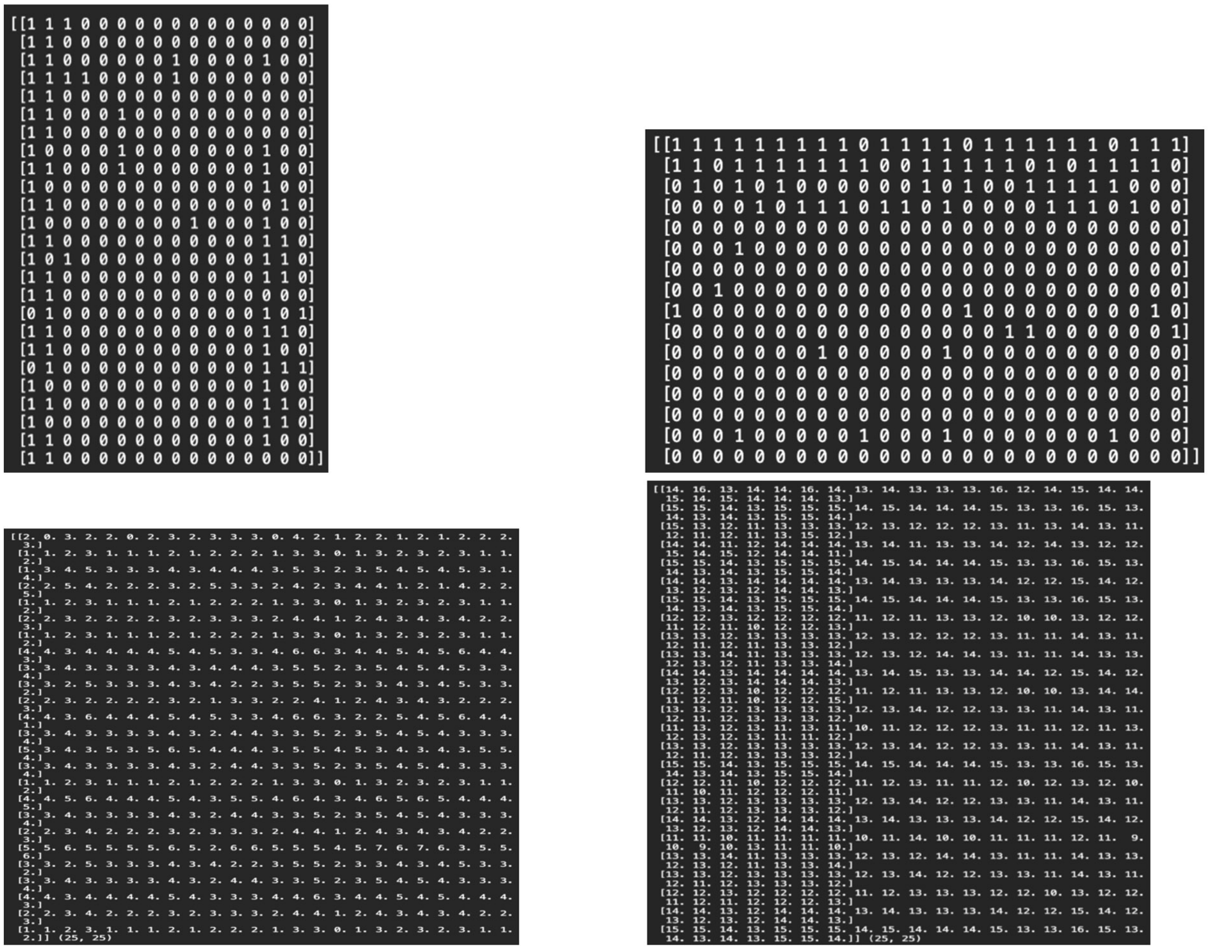
The third method is more like a hybrid of the first two methods. It performs Elementwise XOR weighted sum on the vector matrices giving us the absolute difference of a particular feature vector with all feature vectors in the other vector matrix and vice-versa. This result is a 25 x 25 matrix with correct and true values. This generated matrix is a distance matrix and in-order to get the similarity matrix we must subtract every entry in the distance matrix with 16 which is the total features taken for our problem domain. The high value numbers in the similarity matrix represents the strong closeness and low values represent the least connectivity between the corresponding signs in the similarity matrix.

Finally, we presented these similarity matrices using heat maps for better visualization. We used a color gradient from bright blue to dark red to represent the values inside the matrix where red is assigned for high values and blue for low values.

4 Discussion

In this section we present the feature analysis for the Malayalam Script, screenshots of our process consisting of different matrices we discussed earlier and finally some output heat maps. The heat maps are presented for the Telugu-Malayalam and Kannada-Telugu languages which contains the total of sixteen features in the feature evaluation table.

In the upper left of **Figure 5** from the feature evaluation table all the 16 features for 25 signs are represented in vector notation making it a 25 x 16 matrix, where 25 is the number of signs and 16 is the number of features. Since we need two vector matrices to create a similarity matrix we transpose (upper right of Figure 5) one of the vector matrices to



```
[[1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0]
 [1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
 [1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
 [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

```
[[1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1]
 [1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 1 1 1 1 0]
 [0 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0]
 [0 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 0 1 1 1 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]]
```

```
[[2. 0. 3. 2. 2. 0. 2. 3. 2. 3. 3. 3. 0. 4. 2. 1. 2. 2. 1. 2. 1. 2. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [1. 3. 4. 5. 3. 3. 3. 4. 3. 4. 4. 4. 3. 5. 3. 2. 3. 5. 4. 5. 4. 5. 3. 1.
  4.]
 [2. 2. 5. 4. 2. 2. 2. 3. 2. 5. 3. 3. 2. 4. 2. 3. 4. 4. 1. 2. 1. 4. 2. 2.
  5.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [2. 2. 3. 2. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [4. 4. 3. 4. 4. 4. 4. 5. 4. 5. 3. 3. 4. 6. 6. 3. 4. 4. 5. 4. 5. 6. 4. 4.
  3.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 4. 4. 4. 3. 5. 5. 2. 3. 5. 4. 5. 4. 5. 3. 3.
  4.]
 [3. 3. 2. 5. 3. 3. 3. 4. 3. 4. 2. 2. 3. 5. 5. 2. 3. 3. 4. 3. 4. 5. 3. 3.
  2.]
 [2. 2. 3. 2. 2. 2. 2. 3. 2. 1. 3. 3. 2. 2. 4. 1. 2. 4. 3. 4. 3. 2. 2. 2.
  3.]
 [4. 4. 3. 6. 4. 4. 4. 5. 4. 5. 3. 3. 4. 6. 6. 3. 2. 2. 5. 4. 5. 6. 4. 4.
  1.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [5. 3. 4. 3. 5. 3. 5. 6. 5. 4. 4. 4. 3. 5. 5. 4. 5. 3. 4. 3. 4. 3. 5. 5.
  4.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]
 [4. 4. 5. 6. 4. 4. 4. 5. 4. 3. 5. 5. 4. 6. 4. 3. 4. 6. 5. 6. 5. 4. 4. 4.
  5.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [2. 2. 3. 4. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [5. 5. 6. 5. 5. 5. 5. 6. 5. 2. 6. 6. 5. 5. 5. 4. 5. 7. 6. 7. 6. 3. 5. 5.
  6.]
 [3. 3. 2. 5. 3. 3. 3. 4. 3. 4. 2. 2. 3. 5. 5. 2. 3. 3. 4. 3. 4. 5. 3. 3.
  2.]
 [3. 3. 4. 3. 3. 3. 3. 4. 3. 2. 4. 4. 3. 3. 5. 2. 3. 5. 4. 5. 4. 3. 3. 3.
  4.]
 [4. 4. 3. 4. 4. 4. 4. 5. 4. 3. 3. 3. 4. 4. 6. 3. 4. 4. 5. 4. 5. 4. 4. 4.
  3.]
 [2. 2. 3. 4. 2. 2. 2. 3. 2. 3. 3. 3. 2. 4. 4. 1. 2. 4. 3. 4. 3. 4. 2. 2.
  3.]
 [1. 1. 2. 3. 1. 1. 1. 2. 1. 2. 2. 2. 1. 3. 3. 0. 1. 3. 2. 3. 2. 3. 1. 1.
  2.]] (25, 25)
```

```
[[14. 16. 13. 14. 14. 16. 14. 13. 14. 13. 13. 13. 16. 12. 14. 15. 14. 14.
  15. 14. 15. 14. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [15. 13. 12. 11. 13. 13. 13. 12. 13. 12. 12. 12. 13. 11. 13. 14. 13. 11.
  12. 11. 12. 11. 13. 15. 12.]
 [14. 14. 11. 12. 14. 14. 14. 13. 14. 11. 13. 13. 14. 12. 14. 13. 12. 12.
  15. 14. 15. 12. 14. 14. 11.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [14. 14. 13. 14. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [12. 12. 13. 12. 12. 12. 12. 11. 12. 11. 13. 13. 12. 10. 10. 13. 12. 12.
  11. 12. 11. 10. 12. 12. 13.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 12. 12. 12. 13. 11. 11. 14. 13. 11.
  12. 11. 12. 11. 13. 13. 12.]
 [13. 13. 14. 11. 13. 13. 13. 12. 13. 12. 14. 14. 13. 11. 11. 14. 13. 13.
  12. 13. 12. 11. 13. 13. 14.]
 [14. 14. 13. 14. 14. 14. 14. 13. 14. 15. 13. 13. 14. 14. 12. 15. 14. 12.
  13. 12. 13. 14. 14. 14. 13.]
 [12. 12. 13. 10. 12. 12. 12. 11. 12. 11. 13. 13. 12. 10. 10. 13. 14. 14.
  11. 12. 11. 10. 12. 12. 15.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [11. 13. 12. 13. 11. 13. 11. 10. 11. 12. 12. 12. 13. 11. 11. 12. 11. 13.
  12. 13. 12. 13. 11. 11. 12.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]
 [12. 12. 11. 10. 12. 12. 12. 11. 12. 13. 11. 11. 12. 10. 12. 13. 12. 10.
  11. 10. 11. 12. 12. 12. 11.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [14. 14. 13. 12. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [11. 11. 10. 11. 11. 11. 11. 10. 11. 14. 10. 10. 11. 11. 11. 12. 11.  9.
  10.  9. 10. 13. 11. 11. 10.]
 [13. 13. 14. 11. 13. 13. 13. 12. 13. 12. 14. 14. 13. 11. 11. 14. 13. 13.
  12. 13. 12. 11. 13. 13. 14.]
 [13. 13. 12. 13. 13. 13. 13. 12. 13. 14. 12. 12. 13. 13. 11. 14. 13. 11.
  12. 11. 12. 13. 13. 13. 12.]
 [12. 12. 13. 12. 12. 12. 12. 11. 12. 13. 13. 13. 12. 12. 10. 13. 12. 12.
  11. 12. 11. 12. 12. 12. 13.]
 [14. 14. 13. 12. 14. 14. 14. 13. 14. 13. 13. 13. 14. 12. 12. 15. 14. 12.
  13. 12. 13. 12. 14. 14. 13.]
 [15. 15. 14. 13. 15. 15. 15. 14. 15. 14. 14. 14. 15. 13. 13. 16. 15. 13.
  14. 13. 14. 13. 15. 15. 14.]] (25, 25)
```

Figure 5 Telugu feature matrix (upper left), transpose of the Malayalam feature matrix (upper right), distance matrix (lower left), and a Malayalam-Telugu similarity matrix (lower right).

facilitate the elementwise XOR multiplication. The dot product (25 x 25) of these two matrices, and the XOR matrix do not lead us to the similarity matrix because they perform a simple matrix multiplication and bitwise XOR (25 x 16) respectively. To create a similarity matrix, we need to perform Elementwise XOR multiplication (25 x 25) of the matrices, which calculates the weighted sum of absolute difference between any two feature vectors as shown in the lower left of Figure 5. This is the definition of a distance matrix. The similarity matrix is found by subtracting the total number of features with every element in the 25 x 25 distance matrix as shown in the lower right of Figure 5.

From a similarity matrix, it is easy to generate a heat map. For example, the Malayalam-Telugu heat map is shown in **Figure 6**, and the Kannada-Telugu heat map is shown in **Figure 7**. We can see the highest value of 16 and lowest value of 8 which shows that there are high similar signs and many low similar signs respectively. The graph shows that it is majorly dominated by the red color rather than blue which shows there is a lot of similarity between the two language scripts. Similarly considering the Malayalam and Telugu heat map there are a smaller number of highly matched words which have value of 16 and there is a lot of blue signs in the heat map with lowest value of 9. This shows that both scripts differ a lot compared to the above heat map.

5 Conclusion and Future work

The Dravidian Languages which include Telugu, Tamil, Kannada, and Malayalam are generally known as distinct cousins and are relatively closely related when compared to the Indus Valley Script. Indus valley scripts have been undeciphered until today but there has been a lot of extraction of different kinds of symbols and seals recently. Among the Dravidian languages Telugu and Kannada seem closely related. Though some of the signs in the Tamil script contain a straight-line stroke most of the other signs and signs in other three Dravidian scripts are cursive. This project helps in finding out the similarity between the scripts that are expected to be derived from the undeciphered scripts and help us in finding out the evolution of languages. Our goal is to ease the exhaustive calculations in finding out the similarity matrix between two scripts during comparison. The project has a high scalability factor. It

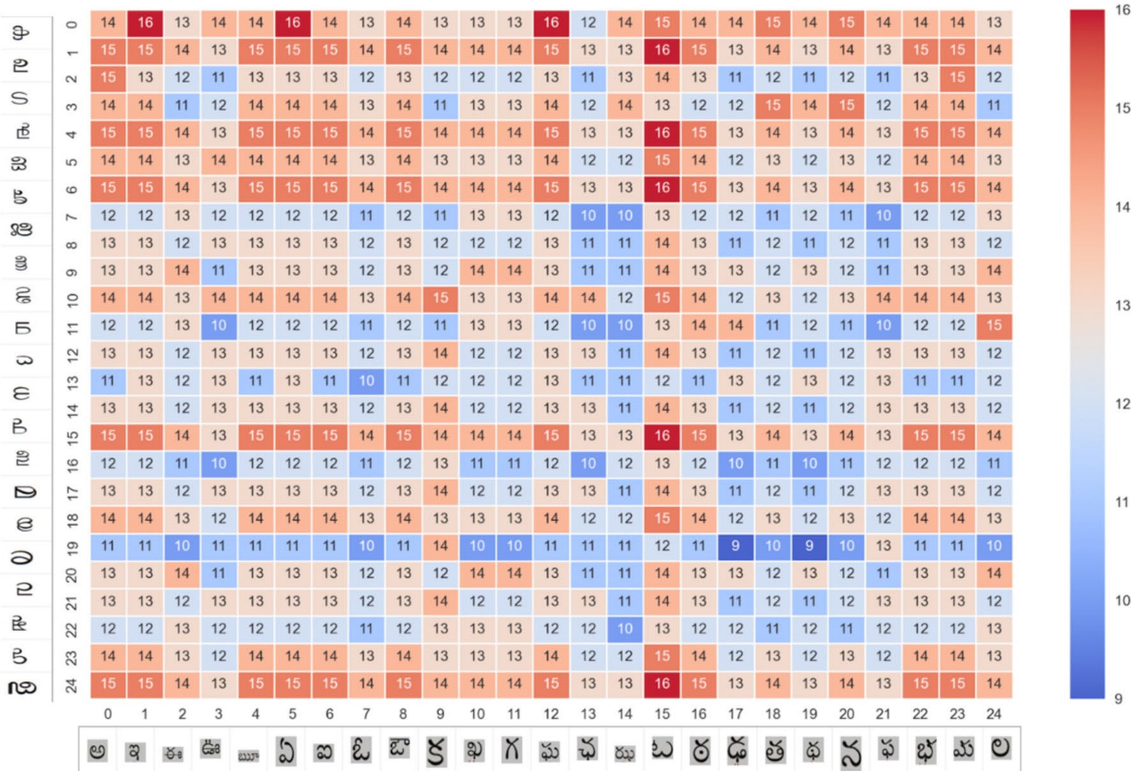


Figure 6 Heat map for Malayalam and Telugu scripts.

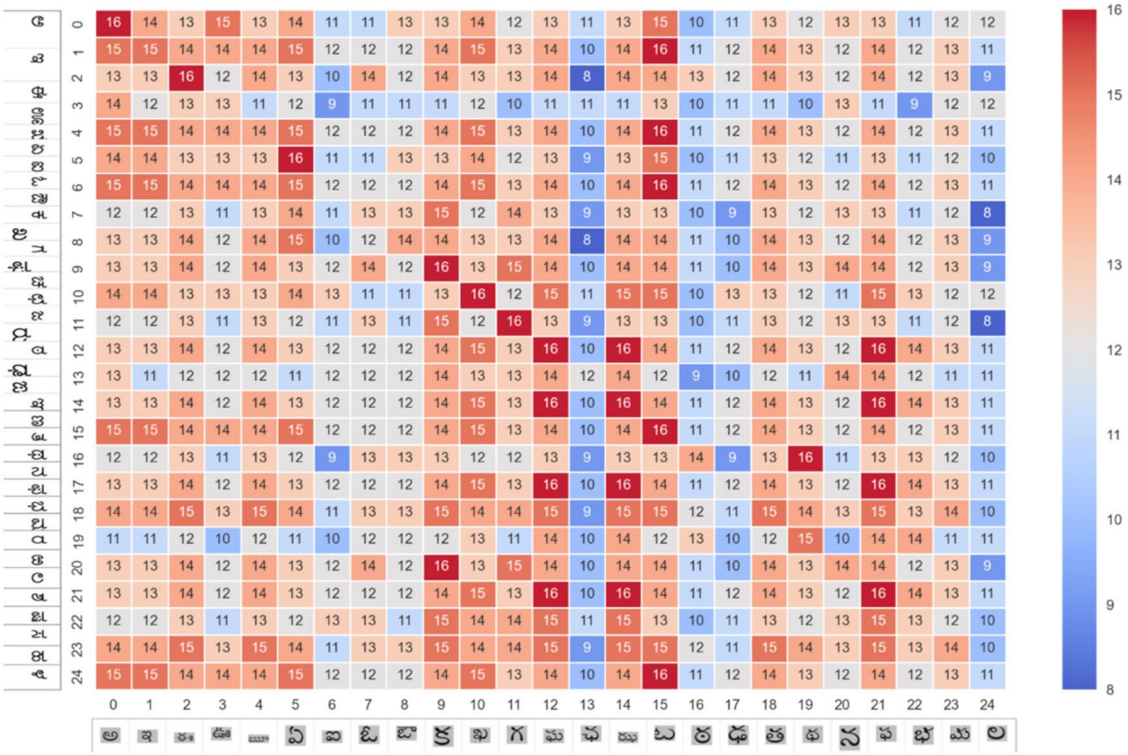


Figure 7 Heat map for Kannada and Telugu scripts.

can be extended by passing the feature vector values directly from the created vector table rather than passing them through NumPy arrays. This process can be flexibly applied to words and thereby construct an evolutionary tree as a future work. In addition, feature analysis can be extended from script analysis to art motif analysis [10] and higher-level textual analysis [11].

References

[1] Shruti Daggumati and Peter Z. Revesz. 2018. Data mining ancient script image data using convolutional neural networks. In Proceedings of the 22nd International Database Engineering and Applications Symposium (IDEAS'18), ACM Press, pp. 209-218. https://doi.org/10.1145/3216122.3216163

[2] Shruti Daggumati and Peter Z. Revesz. 2019. Data mining ancient scripts to investigate their relationships and origins. In Proceedings of the 23rd International Database Engineering and Applications Symposium (IDEAS'19), ACM Press, pp. 209-218. https://doi.org/10.1145/3331076.3331116

[3] Shruti Daggumati and Peter Z. Revesz. 2021. A method of identifying allographs in undeciphered scripts and its application to the Indus Valley Script. Humanities and Social Sciences Communications, 8, 50. https://doi.org/10.1057/s41599-021- 00713-0

[4] Walter Fairservis, Sayid Ghulam Mustafa Shah, and Asko Parpola. 1993. Corpus of Indus Seals and Inscriptions. Vol. 2: Collections in Pakistan. Journal of the American Oriental Society, 113 (2), 310.

[5] Iravatham Mahadevan. 1977. The Indus Script: Texts, concordance and tables, memoirs. Archaeological Survey of India, no. 77.

[6] Asko Parpola, Brij Mohan Pande, and Petteri Koskikallio. 2010. Corpus of Indus Seals and Inscriptions. Vol. 3: New material, untraced objects, and collections outside India and Pakistan, Annales Academiae Scientiarum Fennicae, Humaniora; no. 359.

[7] Jagat Pati Joshi and Asko Parpola. 1987. Corpus of Indus Seals and Inscriptions. Vol. 1: Collections in India. Annales Academiae Scientiarum Fennicae. Series B. no. 239.

[8] Peter Z. Revesz, 2016. Bioinformatics evolutionary tree algorithms reveal the history of the Cretan Script Family. International Journal of Applied Mathematics and Informatics, 10, 67-76.

[9] Peter Z. Revesz, 2017. Establishing the West-Ugric language family with Minoan, Hattic and Hungarian by a decipherment of Linear A. WSEAS Transactions on Information Science and Applications, 14, 306-335.

[10] Peter Z. Revesz, 2019. Art motif similarity measure analysis: Fertile Crescent, Old European, Scythian and Hungarian elements in Minoan culture. WSEAS Transactions on Mathematics, 18, 264-287.

[11] Peter Z. Revesz, 2019. A comparative analysis of Hungarian folk songs and Sanskrit literature using motif similarity matrices. WSEAS Transactions on Information Science and Applications, 16, 75-86.

[12] Abumugam Sathasivam. 1965. Sumerian: A Dravidian Language. Berkeley, California.

[13] Maggie Tallerman. 200. Did our ancestors speak a holistic protolanguage? Lingua, 117 (3), 579-604.

[14] Wikipedia, Dravidian languages. 2022. Available at: https://en.wikipedia.org/wiki/Dravidian_languages