A Simple test of Indo-European Language Family Based on the Edit Distance Function and Hierarchical Clustering

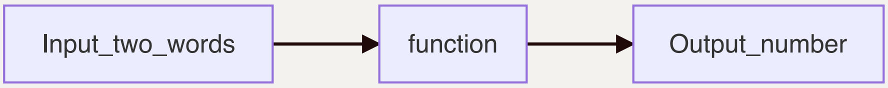
Introduction

The basic idea of analyzing language family is that I could compare the similarity of cognates (i.e: body parts, kinship terms natural phenomena etc.). One way to elucidate the relationship is to use the phylogeny or evolutionary tree. However, it is very difficult for me to mearsure the similarity of word of different languages. In order to construct a tree, I must use the digitalized material. In bioinformatics, scientiest analyze the DNA by a method called DNA Sequence Model, which gives me so much insights because the DNA sequence looks so much like the text.

Digitalizing the data

In order to find a physical quantitier to calculate the similarity, I treated this problem in dynamic way——counting the minimum number of edit operations required to convert one string into the other (i.e I can count how many steps take me to convert one word to another, including adding, deleting and substituting). **Up to here, I wrote a function in Matlab in order to calculate the steps of converting one word to another word, and finally, use a exponential function to output the result**. The following content was how to use this function:

- Choosing an object as a reference
- input two words in the function
- output the number as similarity



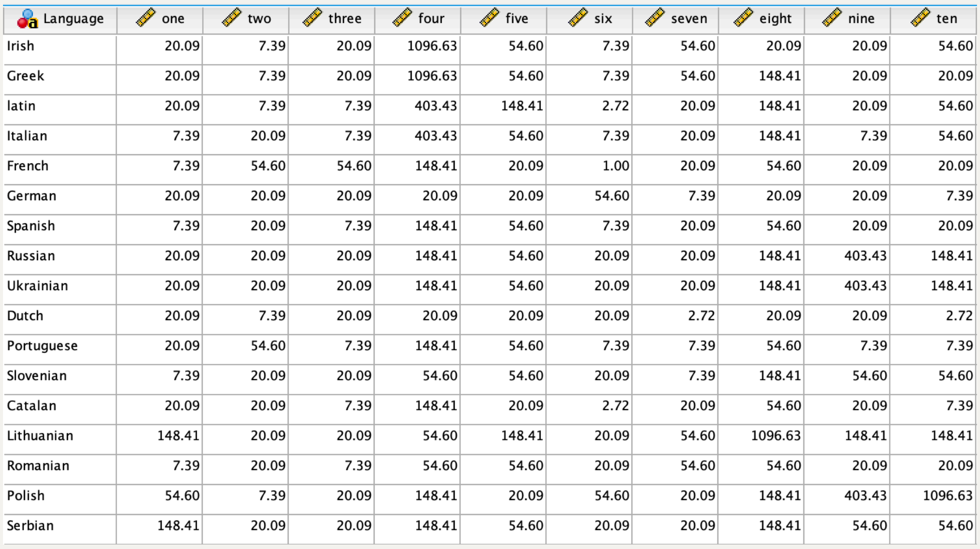
For example, if the step was 0, which implied that the two words were exactly same, so taking 0 into the exponetial function (i.e. exp(0)), which gives 1. Futhermore, The lower similarity, the larger number will output.

Then I chose my experiment object as the expression of number from 1-10 in different languages, comparing the word with Englis, and then used the function I built to digitalize the words. (Since the data type was string, so I use the {} to contain my objects in Matlab).

```
%%The expression of number from 1-10 in different
languages
English=
{'one','two','three','four','five','six','seven','eight'
,'nine','ten'};
Irish=
{'aon','do','tri','ceathair','cuig','se','seacht','ocht'
,'naoi','deich'};
Greek=
{'hen','duo','treis','tettares','pente','hex','hepta','o
kto','ennea','deka'};
Latin=
{'unus','duo','tres','quattuor','quinque','sex','septem'
,'octo','novem','decem'};
Italian=
{'uno','due','tre','quattro','cinque','sei','sette','ott
o','nove','deici'};
French=
{'un','deux','trois','quatre','cinq','six','sept','huit'
,'neuf','dix'};
German=
{'einz','zwei','drei','vier','funf','sechs','sieben','ac
ht','neun','zehn'};
Spanish=
{'uno','dos','tres','cuatro','cinco','seis','seite','och
o','nueve','diez'};
Russian=
{'odin','dva','tri','cetyre','pjat','sest','sem','vosem'
,'devjat','desjat'};
Ukrainian=
{'odin','dva','tri','cetyre','pjat','sest','sem','vosem'
,'devjat','desjat'};
Dutch=
{'een','twee','drie','vier','vijf','zes','zeven','acht',
'negen','tien'};
Portuguese=
{'um','dois','tres','quatro','cinco','seis','sete','oito
','nove','dez'};
```

```
Slovenian=
{'ena','dva','tri','stiri','pet','sest','sedem','osem','
devet','deset'};
Catalan=
{'u','dos','tres','quatre','cinc','sis','set','vuit','no
u','deu'};
Lithuanian{'vienas','du','trys','keturi','penki','sesi',
'septyni','astuoni','deryni','desimt'};
Romanian=
{'unu','doi','trei','patru','cinci','sase','sapte','opt'
,'noua','zece'};
Polish=
{'jeden','dwa','trzy','cztery','piec','szesc','siedem','
osiem','dziewiec','dziesiec'};
Serbian=
{'jedan','dva','tri','cetiri','pet','sest','sedam','osam
','devet','deset'};
```

After digitalizing the array of my objects, I got a matrix and I typed it into Software SPSS:



Language	one	two	three	four	five	six	seven	eight	nine	ten
Irish	20.09	7.39	20.09	1096.63	54.60	7.39	54.60	20.09	20.09	54.60
Greek	20.09	7.39	20.09	1096.63	54.60	7.39	54.60	148.41	20.09	20.09
latin	20.09	7.39	7.39	403.43	148.41	2.72	20.09	148.41	20.09	54.60
Italian	7.39	20.09	7.39	403.43	54.60	7.39	20.09	148.41	7.39	54.60
French	7.39	54.60	54.60	148.41	20.09	1.00	20.09	54.60	20.09	20.09
German	20.09	20.09	20.09	20.09	20.09	54.60	7.39	20.09	20.09	7.39
Spanish	7.39	20.09	7.39	148.41	54.60	7.39	20.09	54.60	20.09	20.09
Russian	20.09	20.09	20.09	148.41	54.60	20.09	20.09	148.41	403.43	148.41
Ukrainian	20.09	20.09	20.09	148.41	54.60	20.09	20.09	148.41	403.43	148.41
Dutch	20.09	7.39	20.09	20.09	20.09	20.09	2.72	20.09	20.09	2.72
Portuguese	20.09	54.60	7.39	148.41	54.60	7.39	7.39	54.60	7.39	7.39
Slovenian	7.39	20.09	20.09	54.60	54.60	20.09	7.39	148.41	54.60	54.60
Catalan	20.09	20.09	7.39	148.41	20.09	2.72	20.09	54.60	20.09	7.39
Lithuanian	148.41	20.09	20.09	54.60	148.41	20.09	54.60	1096.63	148.41	148.41
Romanian	7.39	20.09	7.39	54.60	54.60	20.09	54.60	54.60	20.09	20.09
Polish	54.60	7.39	20.09	148.41	20.09	54.60	20.09	148.41	403.43	1096.63
Serbian	148.41	20.09	20.09	148.41	54.60	20.09	20.09	148.41	54.60	54.60

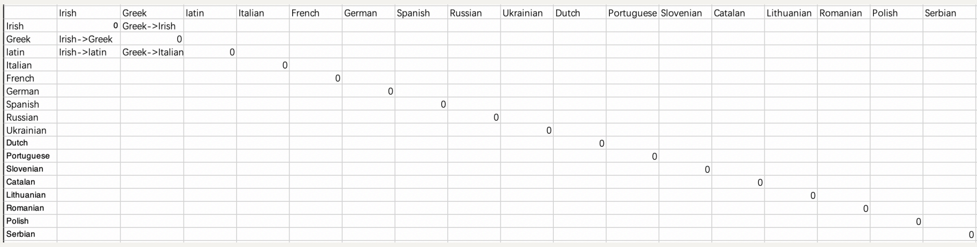
In the matrix, it was very clear to identifiy the similarity of the cognates, eg: the number 6 in French was six, which was the same as English, so the digitalized data was 1. Futhermore, the larger number implied lower similarity.

Hierarchical Clustering

I used the hierarchial Clustering to model the relationship of different languages. **The main idea of hierarchical clustering was by comparing the distance to divide the space into serveral classifications** .

Steps:

- Forming a matrix by puting the object on the column and row
- Choosing a distance: such as EuclideanDistance, Standardized Euclidean distance, cosine distance, Mahalanobis distance, Chebyshev Distance etc.
- Calculating the distance
- Selecting the minimum distance and combine them together
- Using the combined data as new object
- Iterating



	Irish	Greek	latin	Italian	French	German	Spanish	Russian	Ukrainian	Dutch	Portuguese	Slovenian	Catalan	Lithuanian	Romanian	Polish	Serbian
Irish	0	Greek->Irish															
Greek	Irish->Greek	0															
latin	Irish->latin	Greek->Italian	0														
Italian				0													
French					0												
German						0											
Spanish							0										
Russian								0									
Ukrainian									0								
Dutch										0							
Portuguese											0						
Slovenian												0					
Catalan													0				
Lithuanian														0			
Romanian															0		
Polish																0	
Serbian																	0

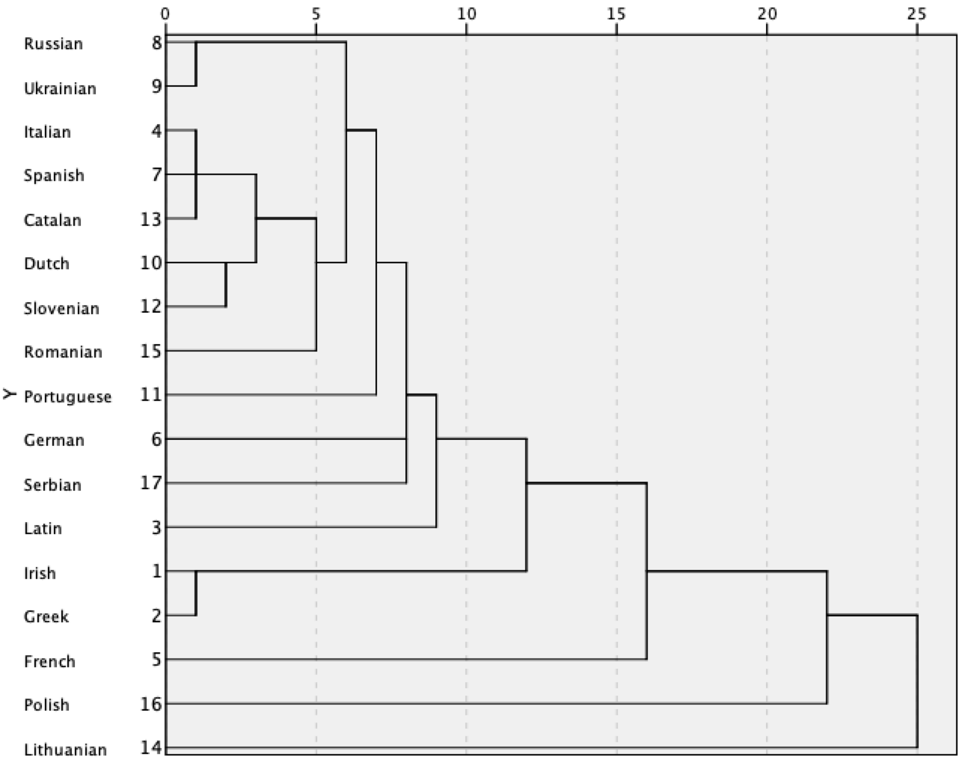
For distance, I use the **Euclid space distance**, which was calculated by the following formula:

$$D_{A->B} = \sqrt{(A-B)^2}$$

Since the value was always positive, so the matrix would be symmetric along the diagonal.

Then I choose the smallest distance, which means high similarity, and combine them together. In my experiment, for example, the Russian and Ukranian would be combined together as a new number (Russian, Ukranian), which fomed the most bottom of the tree.

Iterating this step, and ploted them by the Software SPSS, I got this tree:



Since my sample is very small, it could not imply the whole Indo-European Language family, but what I want to do was to test whether I could use this method. Based on the common sense or compared it with the tree I searched on the Internet, the result in my experiment was to some extend reasonable.

Problems

- The sample in this test is too small
- The converting function might be too simple
- Different distance type gives different results, since the choosing process is too subjective, so it is very difficult to identify which type of distance is the best one
- The result might be more accurate by increasing the words expression.
- It is necessary to choose a type of language as reference, which means that Enghlish could not be included in the result.