Project Work :

UE15CS490(Major)/UE15CS492(Minor)

Final ISA(Review 5) / ESA 2019

Project Title : Linguistic Analysis of Indo-European Languages
Project ID : **PW19SMP003**
Project Guide : Prof. Shreekanth M Prabhu
Project Team : Roshan U[01FB15ECS246],
Sanath Bhimsen[01FB15ECS260],
Mukesh M Karanth[01FB15ECS361].

Problem Statement

- We thought that the current model of Indo-European language evolution theory might be a biased model due to the limited considerations and the restricted visualisation of the languages.
- Therefore, We want to include possibilities of word transfers and mutual growth, resulting in the growth of both the languages involved.
- After using these ideas and considerations, we want to come up with a better, more realistic visualization of the Indo-European Languages, one that may resemble an network with various links and crosslinks that connect languages that were thought to have developed independently.

Literature Survey

[1] Renfrew, Colin. "The Origins of Indo-European Languages." Scientific American 261, no. 4 (1989): 106-15. http://www.jstor.org/stable/24987446.

[2] Boc, Alix, Anna Maria Di Sciullo, and Vladimir Makarenkov. "Classification of the Indo-European languages using a phylogenetic network approach." In Classification as a Tool for Research, pp. 647-655. Springer, Berlin, Heidelberg, 2010.

[3] Prabhu, Shreekanth. (2018). Evolving a Framework to interpret the Vedas. 10.13140/RG.2.2.32939.95529.

[4]. Clustering Semantically Equivalent Words into Cognate Sets in Multilingual Lists http://www.aclweb.org/anthology/I11-1097

[5]. Mapping the Origins and Expansion of the Indo-European Language Family Remco Bouckaert, Philippe Lemey, Science 24 Aug 2012:Vol. 337, Issue 6097, pp. 957-960 DOI: 10.1126/science.1219669

[6]. The Origins of Indo-European Languages Colin Renfrew Scientific American Vol. 261, No. 4 (OCTOBER 1989), pp. 106-115
Dept.

Proposed Solution

○ Collect the dataset of Indo-European languages from Langfocus website and other similar websites.

○ Select a few key languages and pre-process the dataset for any discrepancies.

○ Perform analysis on the dataset, by getting distance between languages by the closeness of their words using distance measures like Levenshtein distance, etc. and centrality measures.

- Understand and apply Horizontal Gene Transfer Detection Algorithm.

- Combine all the results and visualise the dataset to obtain a new and better model of the layout of Indo-European Languages.

Why Your Solution is Better?

Features of the solution

- Consideration of other possibilities like Mutual growth and word sharing.
- Adapting a new and unique approach of dynamic dataset building that takes synonyms of the word closest to the transliterated words.
- Visualization in the form of a network to depict linkages between independent languages.

Technologies / Methodologies

TECHNOLOGIES:

-> Applications:

-Google Translate API: Used to retrieve transliterated words quickly and in bulk.

-Jupyter Notebook: For its GUI and ease of use.

-Rstudio: consolidated representation of results, command prompt and single shot execution of programs.

-NodeXL: It provides an easy to use GUI with simple drag & drop options and a multitude of similarity measures at the click of a button, excellent for visualization.

-> Languages:

- Python: Used because of its ease of programing and amazing libraries that provide wide range of functionality in analytics.
- R : Used for visualisation because of its packages that aid good visualisation and its simplicity.

PES UNIVERSITY

Dependencies and Risks

DEPENDENCIES:

-> The usage is strictly limited only to study the effects and results of social network analysis on Linguistics and its representations.

-> The project requires packages which are good for visualization purposes and have good functionality for Network Analysis.

ASSUMPTIONS:

->A single language was taken from each of the lineages of the Indo-European Language, Proto-Indo-European.

->The number of words chosen to represent each language were all transliterated in English.

-> The maximum number of words in a single language is restricted to 200.

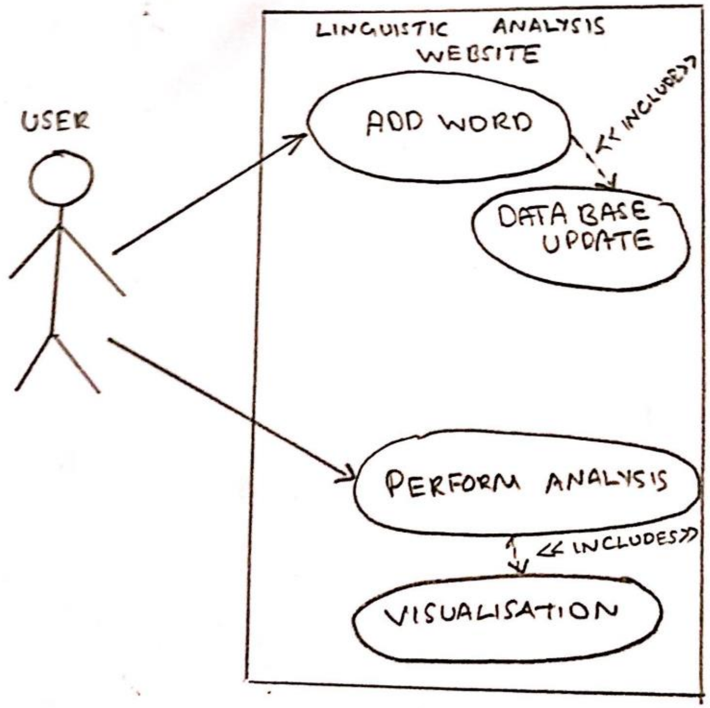
RISKS:

-> The limited number of centrality and similarity measures used for analysis could potentially affect the accuracy of the visualisation.

-> The limited pool of words chosen might not be sufficient enough to convey any good results.

-> The words chosen might not be the best choice to best depict all languages perfectly.

USE CASE DIAGRAM



Modules

1) **Collections of English Words** : In this module, We started off by collecting around 1000 English words and then filtered it down to 300 words suitable words for our study. We made sure that the data had a good number of stop words, nouns, proverbs, adjectives and words representing relationships like Father, Mother, Etc.

2) **Translation of English Words** : Once this was done , The googletrans python API was used in order to translate each english words into the other Indo-European Languages we had chosen. They are German , Italian , French , Spanish , Hindi ,Russian, Latin and Sanskrit. Since there is no support for Sanskrit on Google Translate , We used an English to Sanskrit Dictionary in order to find the Sanskrit Representation of the word.

3) **Loading all these words into a dataframe** : The dataset is prepared this way and it is stored in csv format columns represent the languages and we have the words and their representation on each row.

4) **Preprocessing** : Once the dataset was created, each word was replaced with its phonetic pronunciation using manual or automatic conversion wherever applicable. As the API might sometimes return the word in its native representation, this would negatively affect our results because the special characters present as a translation of certain words in other languages will not be considered close to a word even if it sounds the same when pronounced, this is because of the way the similarity measure is computed.

5) **Weighted Similarity Graph Using NetworkX** : The python library networkX was used to prepare a visualization that shows how close every pair of languages are. We used weighted edges for the same. In this module, each word of every language taken for the study for a particular meaning will be taken and compared with each other and their similarity score will be computed for every possible pair and the result will be stored and returned as a matrix.

6) **Creation of Distance Matrix** : For every english word we construct distance matrix showing the distance between that word and its representation in all other languages.
Since we have nine languages in total we will create a 9x9 matrix that stores the distance matrix for all language’s words for a particular meaning. This will be used for the detection of clusters, communities and hidden linkages between the words from different languages. This will give us a basic idea of the potential relationship that might exist between two previously independent languages.

7) **Cognate Cluster Creation** : The distance matrix for each word is passed to the lingpy cognate cluster creation method and it returns a dictionary. The keys of the dictionary represent the cluster number and the value is a list which contains the word that falls in that cluster. These results are stored in a csv file and passed as an input for the visualization modules.

8) **Network Nodes Creation** : The cognate clusters stored in the csv file are read into a dataframe variable in R and are converted into a format readable by the network node creator function. That is, the cluster number and the fused attribute of word and language are passed as the two parameters for the node creation function, the function then creates a base network layout of all the nodes.

9) **Visualization in R** : The network layout is then passed to the community detector which portrays the nodes as individually highlighted clusters of different colors, which give us a good idea of the hidden relationships that may potentially exist between two independent languages.

Design approach: **Top-Down Approach**

Benefits:

- **Decreased Risk:** Since the approach is planned well in advance.
- **Good Organization:** Tasks are determined and filtered down without any confusion because project goals are set and will not be affected by outside opinions.
- **Minimized Cost:** Members are free to complete their own tasks unique to their role in the project and aren't saddled with the responsibility of setting project-wide goals.

Drawbacks:

- Limited Creativity: Members are engrossed in their responsibilities and are unable to contribute innovations/ideas to the project — sometimes leading to frustration and a lack of motivation to perform.
- Slow Response to Challenges: When a challenge arises as a result of a decision, it can take time for the members to establish a solution because there are limited minds contributing to decisions.

Testing Methodologies:

Data set:

- Check the words in the dataset against the words in the google grammar for the language to validate it's existence.
- Compare the phonetically translated word against its actual phonetic pronunciation to cross check the valid translation.
- Check for alternate forms of the same word.

Code Output:

- Check the output for multiple data sets.
- Perform analysis on different types of words and compare the results.
- Visual Inspection of outputs to verify the code.
- Compare the visual tree with the online sources to validate output.
- Validate the outputs observed using domain knowledge.

Project Completion

Project Report Status – Ready

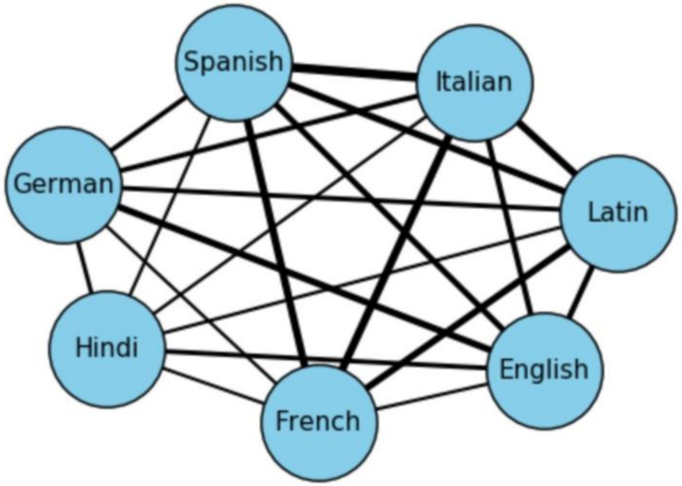
- Page count: 55
- Report type: research report

Project Demo – Fully done

- complete in almost all respects
- working of the project
- Data set creation needs to be demoed

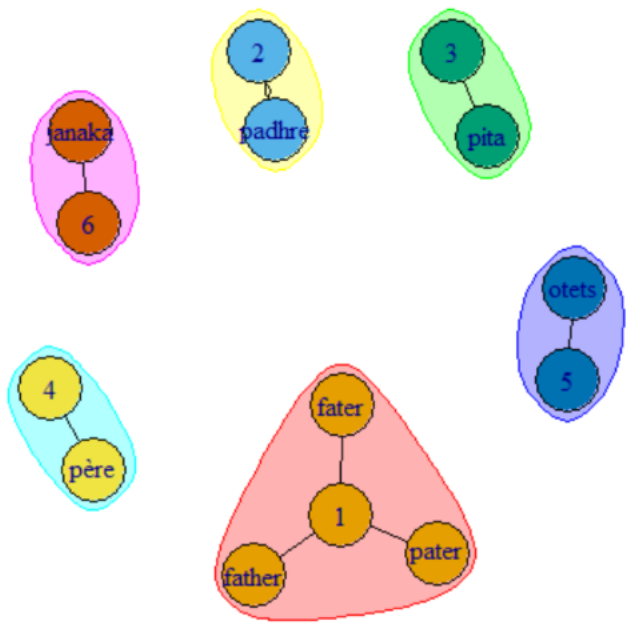
Project Results

Weighted Graph Showing Similarities between Languages



```
{1: ['father', 'fater', 'pater'],
 2: ['père', 'padhre', 'padhre'],
 3: ['pita'],
 4: ['otets'],
 5: ['janaka']}
```

Cluster	Language	Word
1	English	father
1	German	fater
1	Latin	pater
2	Italian	padhre
2	Spanish	padhre
3	Hindi	pita
4	French	père
5	Russian	otets
6	Sanskrit	janaka

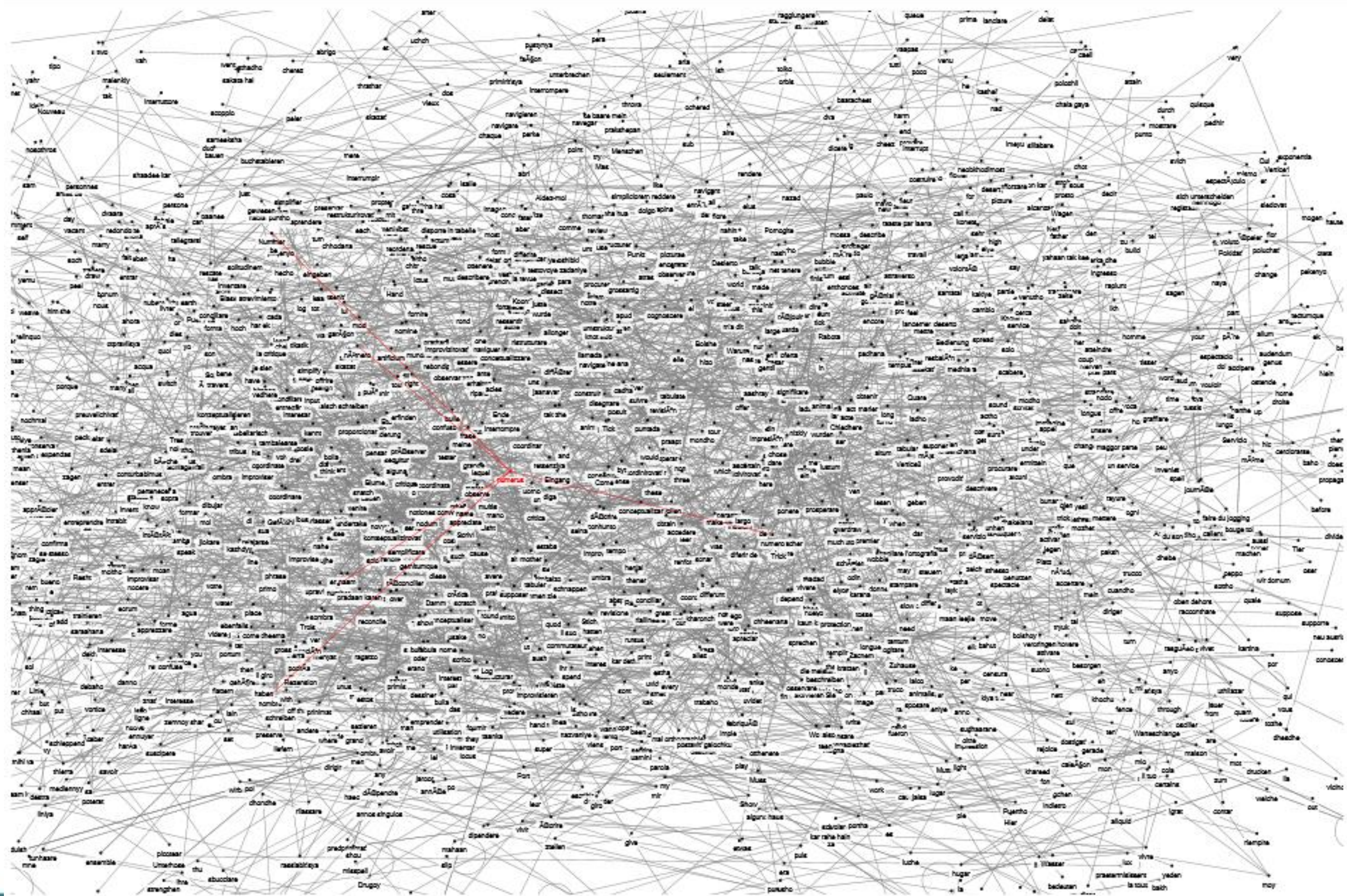


Clusters for the word “Father”

	A	B	C	D	E
1		Vertice1	Vertice2	Link_Strength	Score
2	0	of	fon	Strong	0.4
3	1	in	im	Strong	0.5
4	2	is	is	Strong	1
5	3	it	est	Strong	0.4
6	4	he	er	Strong	0.5
7	5	was	waar	Strong	0.5714285714
8	6	with	mit	Strong	0.5714285714
9	7	have	haben	Strong	0.6666666667
10	8	this	diese	Strong	0.4444444444
11	9	from	fon	Strong	0.5714285714
12	10	or	oder	Strong	0.6666666667
13	11	had	hatten	Strong	0.4444444444
14	12	word	Wort	Strong	0.5
15	13	we	wir	Strong	0.4
16	14	can	kannen	Strong	0.4444444444
17	15	other	andere	Strong	0.3636363636
18	16	were	wurden	Strong	0.4
19	17	all	allez	Strong	0.75
20	18	when	wann	Strong	0.5
21	19	use	benutzen	Strong	0.3636363636
22	20	an	ein	Strong	0.4
23	21	which	welche	Strong	0.5454545455
24	22	their	ihr	Strong	0.5
25	23	write	schreiben	Strong	0.4285714286
26	24	would	wurde	Strong	0.6
27	25	so	so	Strong	1

Graph Metric	Value
Graph Type	Undirected
Vertices	1658
Unique Edges	1995
Edges With Duplicates	561
Total Edges	2556
Self-Loops	103
Reciprocated Vertex Pair Ratio	Not Applicable
Reciprocated Edge Ratio	Not Applicable
Connected Components	321
Single-Vertex Connected Components	11
Maximum Vertices in a Connected Component	69
Maximum Edges in a Connected Component	136
Maximum Geodesic Distance (Diameter)	15
Average Geodesic Distance	3.104348
Graph Density	0.001579729
Modularity	Not Applicable
NodeXL Version	1.0.1.381

Readability Metric	Value



```
In [20]: cd_final

Out[20]: {'English': 2.506666666666667,
 'French': 2.14,
 'German': 1.6266666666666667,
 'Hindi': 1.1733333333333333,
 'Italian': 2.6566666666666667,
 'Latin': 2.3833333333333333,
 'Russian': 1.3166666666666667,
 'Spanish': 2.5366666666666666}
```

We can see that italian is the most central language from the above degree centrality computations

Planned Effort Vs Actual Effort

Planned Effort

Week Number	Project Phase	Responsibility
1 - 3	Understanding and Research on project	All Members
3	Dataset Collection	Roshan
4 - 6	Similarity Computations and Cognate formation	Sanath
7	Visualization	Mukesh
8	Integration and Testing	All Members

Actual Effort

Role	Time taken	Responsibility
Research on project	3 Weeks	All Members
Dataset Collection	4 Weeks	Sanath
Data Pre-Processing	3 Weeks	Roshan
Similarity & Cognates	2 Weeks	Sanath, Mukesh
Visualization	3 Weeks	NodeXL: Sanath, Roshan R : Mukesh
Documentation	15 Weeks	Mukesh
Final Project Report	1 Week	All Members

Lessons Learnt

- This project has helped us in understanding the linguistic analysis domain, it has given us a better idea about the various challenges and methodologies that are involved in a successful attempt at linguistic analysis
- In our attempt at performing Linguistic analysis on the Indo-European languages and their evolution and development theory, we have come across a new approach on expanding and building up a database from scratch which will give a better quality dataset to work on. This approach is more efficient and time saving because of its semi-automated nature.
- Finally, we obtained new insights and visualizations that have helped us depict the Indo-European languages as a network like structure within which the languages have links between themselves which was not thought to have existed in the original tree like structure.

Further Plans for the Fututre

- Improve the NodeXL visualization to filter the nodes further to get a clear idea of the underlying network structure.
- Infer further details about the visualization like most commonly spread word, most central words in all languages, etc.
- Improve the NodeXL visualization into R or Python and then colour code the nodes to understand the linkages between nodes better.

Thank
You