Project Progress Review #4
(Implementation & Testing)

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].

Project Abstract and Scope

This project is a research oriented project which deals with linguistic analysis of Indo-European Languages using Social Network analysis.

We use data set that contains words from multiple languages to perform similarity measures and centrality measures between the words of different languages to find hidden links between languages.

The scope of the project is subject to the project being a minor project with time constraints, hence we are making use of transliterated words, the number of words is limited to a max of 200 and we are using a select number of centrality and similarity measures.

Panel Modifications and Requirements

Panel Requirements:-

- Use Phonetic Pronunciations of words instead of plain transliterated words.
- Consider Nouns, Verbs and other relationship oriented words instead of prepositions, adjectives, etc.
- Use Russian language instead of the Persian language.
- Use a good data corpus of around 100+ words and 5+ languages.
- Hyper-graph visualization of the words.

Project Testing and Implementation

Testing Methodologies:

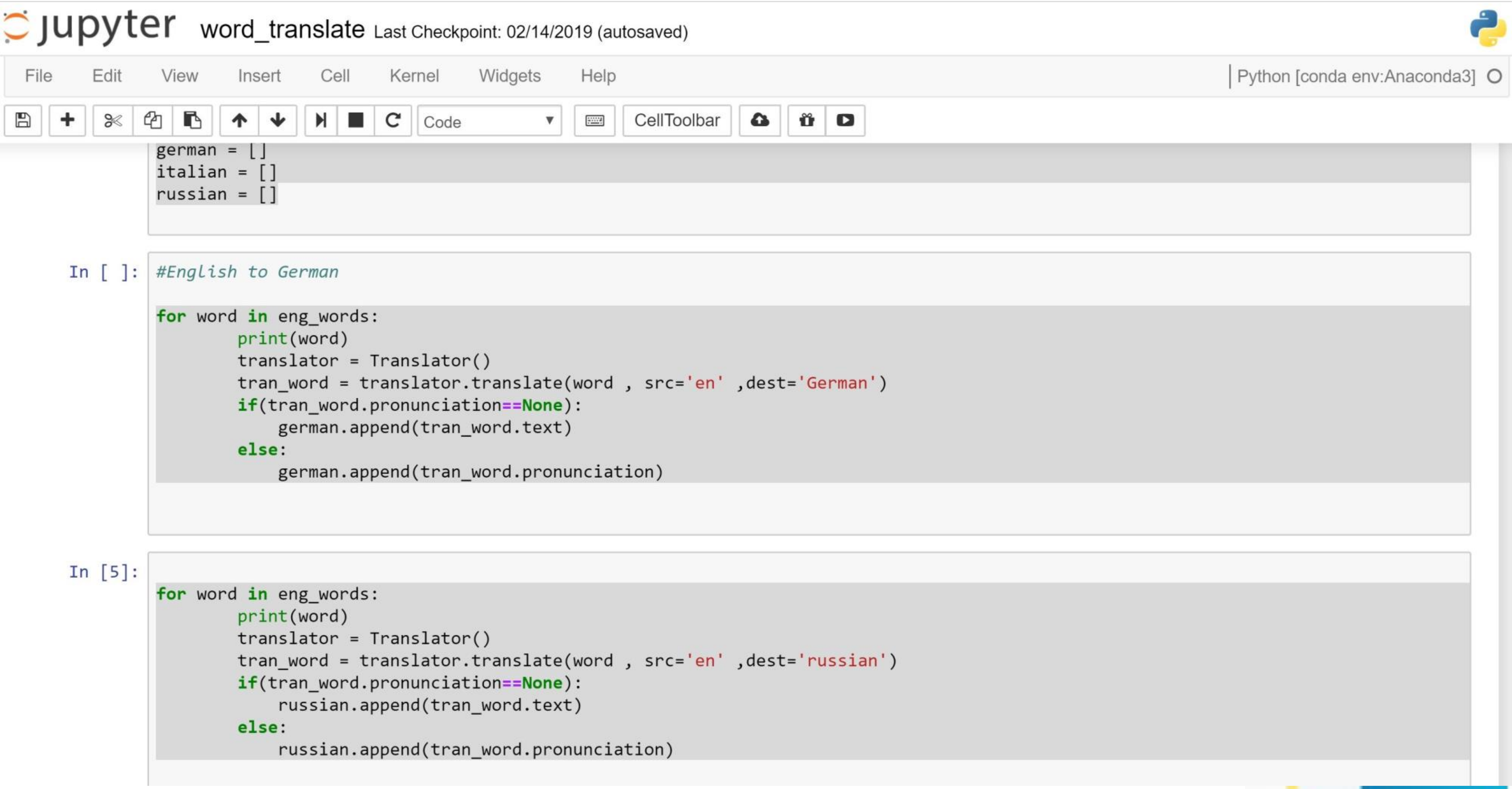
Data set:

- 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.

Implementation



jupyter
word_translate Last Checkpoint: 02/14/2019 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python [conda env:Anaconda3]
Code
CellToolbar

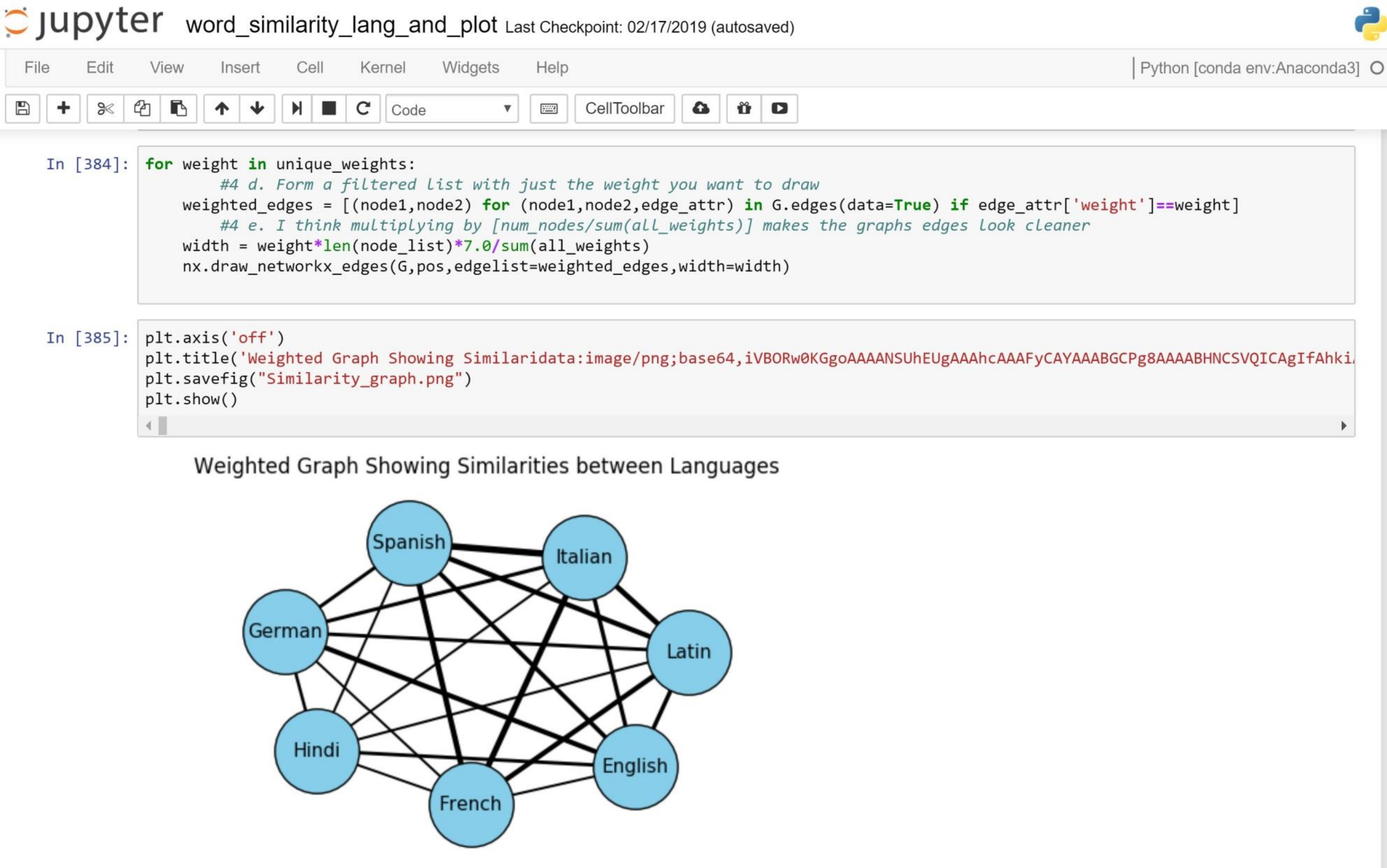
```
german = []
italian = []
russian = []
```

In [ ]:

```
#English to German

for word in eng_words:
        print(word)
        translator = Translator()
        tran_word = translator.translate(word , src='en' ,dest='German')
        if(tran_word.pronunciation==None):
            german.append(tran_word.text)
        else:
            german.append(tran_word.pronunciation)
```

In [5]:

```
for word in eng_words:
        print(word)
        translator = Translator()
        tran_word = translator.translate(word , src='en' ,dest='russian')
        if(tran_word.pronunciation==None):
            russian.append(tran_word.text)
        else:
            russian.append(tran_word.pronunciation)
```

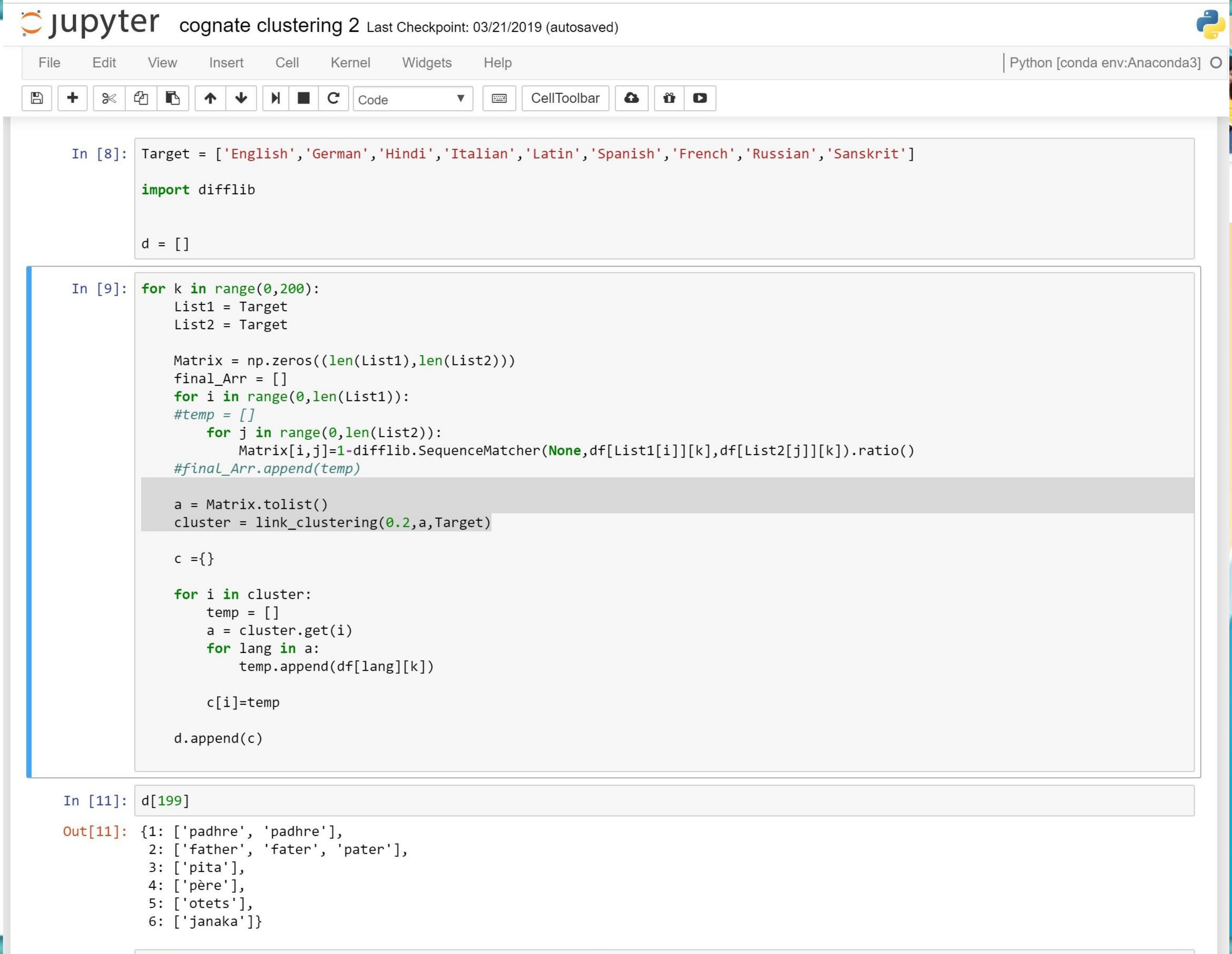


word_similarity_lang_and_plot Last Checkpoint: 02/17/2019 (autosaved)

```
In [384]: for weight in unique_weights:
                #4 d. Form a filtered list with just the weight you want to draw
            weighted_edges = [(node1,node2) for (node1,node2,edge_attr) in G.edges(data=True) if edge_attr['weight']==weight]
                #4 e. I think multiplying by [num_nodes/sum(all_weights)] makes the graphs edges look cleaner
            width = weight*len(node_list)*7.0/sum(all_weights)
            nx.draw_networkx_edges(G,pos,edgelist=weighted_edges,width=width)
```

```
In [385]: plt.axis('off')
          plt.title('Weighted Graph Showing Similaridata:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAhcAAAFyCAYAAABGCPg8AAAABHNCSVQICAgIfAhki
          plt.savefig("Similarity_graph.png")
          plt.show()
```

Weighted Graph Showing Similarities between Languages


jupyter
cognate clustering 2 Last Checkpoint: 03/21/2019 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python [conda env:Anaconda3]
Code
CellToolbar

In [8]:

```
Target = ['English','German','Hindi','Italian','Latin','Spanish','French','Russian','Sanskrit']

import difflib


d = []
```

In [9]:

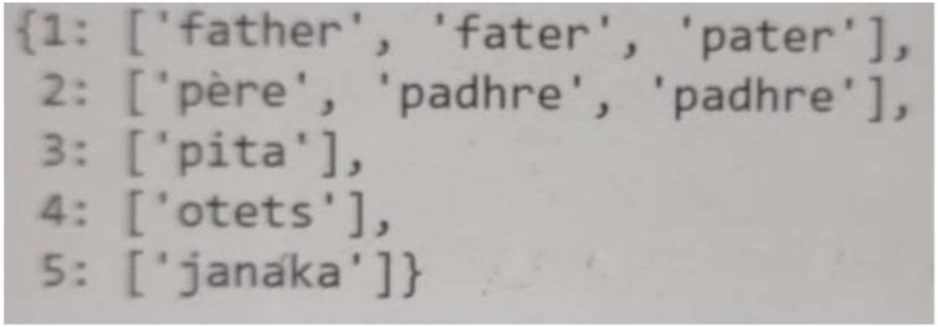
```
for k in range(0,200):
    List1 = Target
    List2 = Target

    Matrix = np.zeros((len(List1),len(List2)))
    final_Arr = []
    for i in range(0,len(List1)):
    #temp = []
        for j in range(0,len(List2)):
            Matrix[i,j]=1-difflib.SequenceMatcher(None,df[List1[i]][k],df[List2[j]][k]).ratio()
    #final_Arr.append(temp)

    a = Matrix.tolist()
    cluster = link_clustering(0.2,a,Target)

    c ={}

    for i in cluster:
        temp = []
        a = cluster.get(i)
        for lang in a:
            temp.append(df[lang][k])

        c[i]=temp

    d.append(c)
```

In [11]:

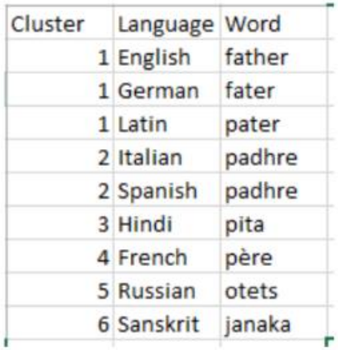
```
d[199]
```

Out[11]:

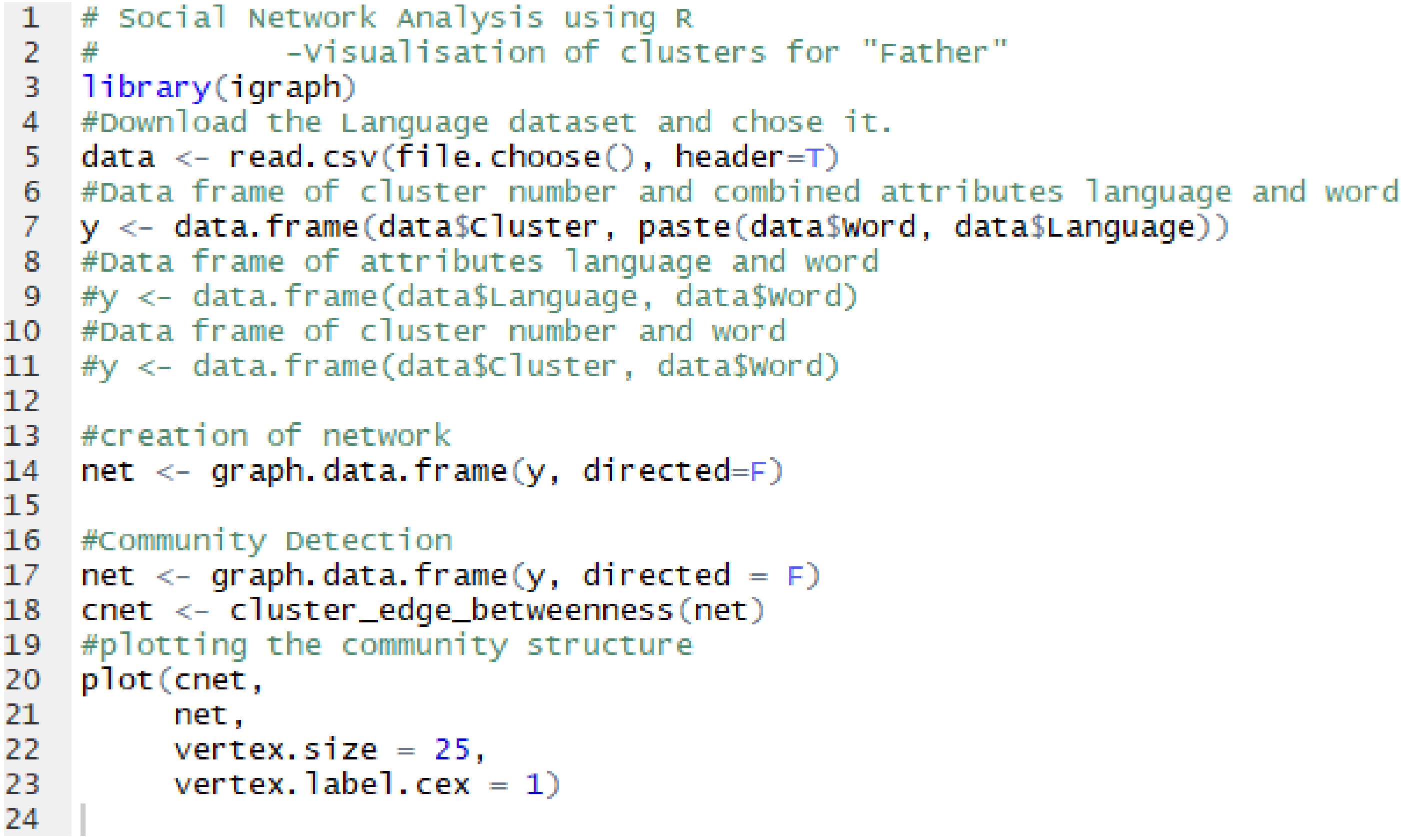
```
{1: ['padhre', 'padhre'],
 2: ['father', 'fater', 'pater'],
 3: ['pita'],
 4: ['père'],
 5: ['otets'],
 6: ['janaka']}
```



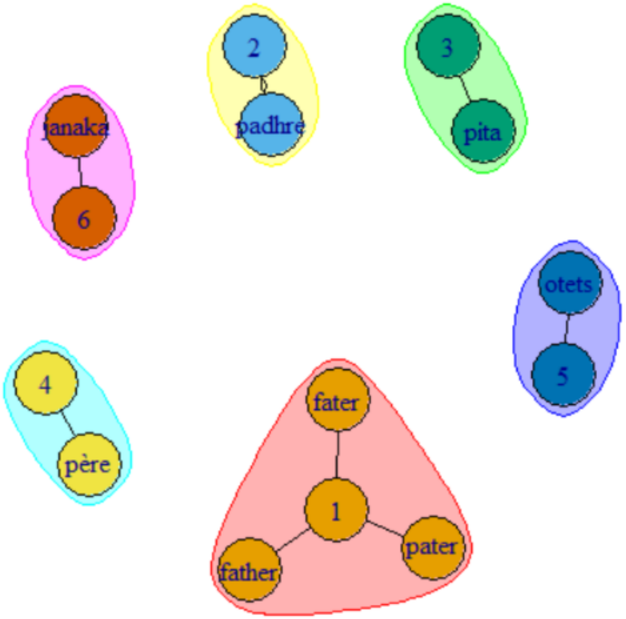
```
{1: ['father', 'fater', 'pater'],
 2: ['père', 'padhre', 'padhre'],
 3: ['pita'],
 4: ['otets'],
 5: ['janaka']}
```



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



```r
# Social Network Analysis using R
#          -Visualisation of clusters for "Father"
library(igraph)
#Download the Language dataset and chose it.
data <- read.csv(file.choose(), header=T)
#Data frame of cluster number and combined attributes language and word
y <- data.frame(data$Cluster, paste(data$Word, data$Language))
#Data frame of attributes language and word
#y <- data.frame(data$Language, data$Word)
#Data frame of cluster number and word
#y <- data.frame(data$Cluster, data$Word)

#creation of network
net <- graph.data.frame(y, directed=F)

#Community Detection
net <- graph.data.frame(y, directed = F)
cnet <- cluster_edge_betweenness(net)
#plotting the community structure
plot(cnet,
     net,
     vertex.size = 25,
     vertex.label.cex = 1)

```



Clusters for the word “Father”

Project Progress So far

What is the project progress so far?

- The Project is near completion and we have incorporated many of the panel requirements into the project.
- We have appended 100 more words which are nouns and verbs specifically.
- Performed Origin tree analysis on the words in the dataset.
- Visualized the clustering of words in accordance to their similarities.

Status of documentation

- CRS or equivalent for research projects: Completed
- HLD/LLD or equivalent for research projects: Completed
- Test Strategy / Test Plan documents: Pending
- Final Project Report Status : Pending

Project Demo

Demo of the project:

- Showing working of the project.
- Showing Origin Tree visualization.
- Data set creation needs will be demoed as usual.

Thank You