Karthik B 19BAI1076

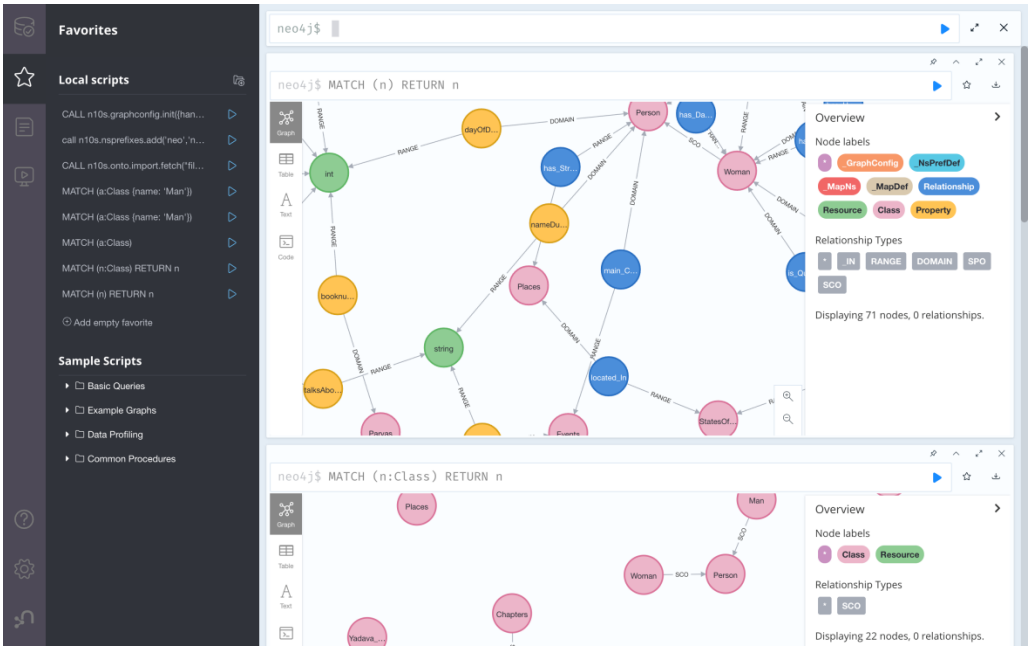
Bharathwaj Murali 19BAI1137

A Simple and Efficient Framework for Identifying Relation-gaps in Ontologies

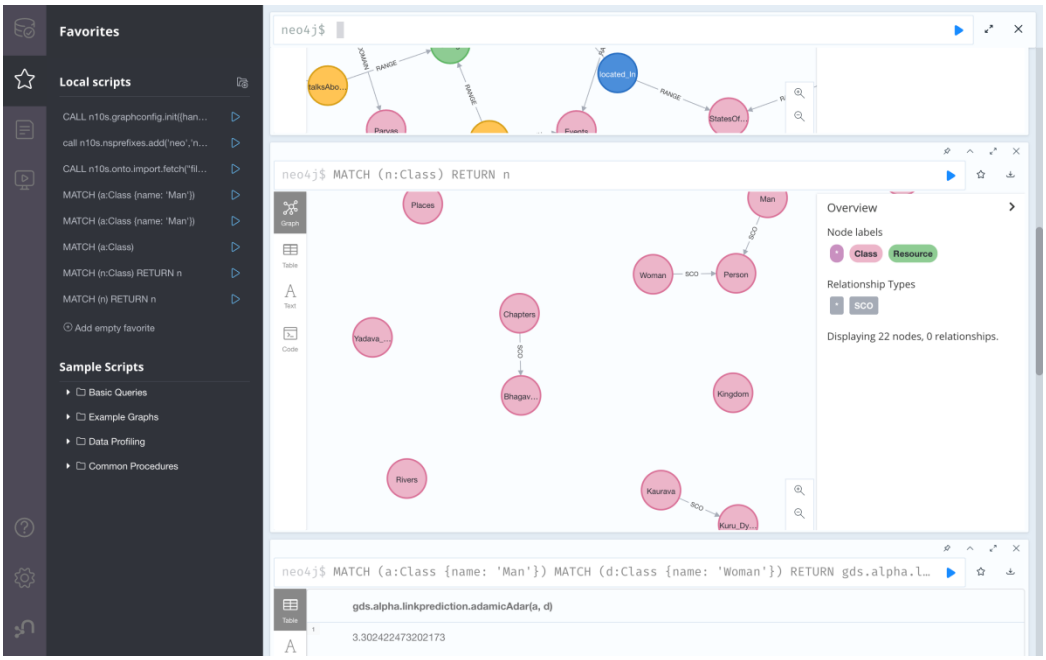
Abstract:

The goal of this project is to find potential pairs of classes that could be connected by an object property but have not yet been connected. Such pairs of classes are called relation-gaps. Relation-Gaps are termed as a link prediction task that is used in graph-based data models, to predict edges that will appear in near future. We propose a simple machine learning framework which exhibits low time complexity and yet gives promising results with respect to both precision as well as number of class-pairs retrieved. This includes a methodology to predict potential pairs of classes which could be connected by object properties but are not yet connected. We take the evidence obtained from external textual resources and their Word2Vec representations can be made use of, for this purpose. Our approach gives results that are complementary to those given by the traditional techniques found in the literature. Hence our method can be used in combination with the traditional techniques for maximum benefits.

Mahabharata Ontology



Classes



Measure of Proximity or Similarity between two nodes

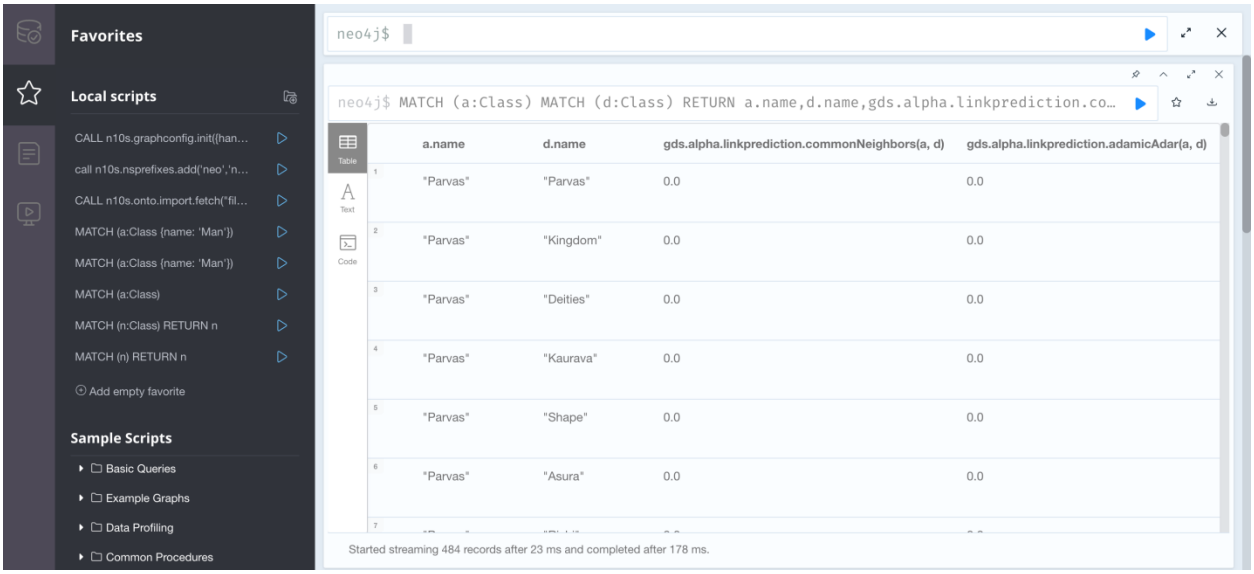
Common Neighbors:

As the name suggests, this measure computes the number of common neighbors that a pair of nodes share.

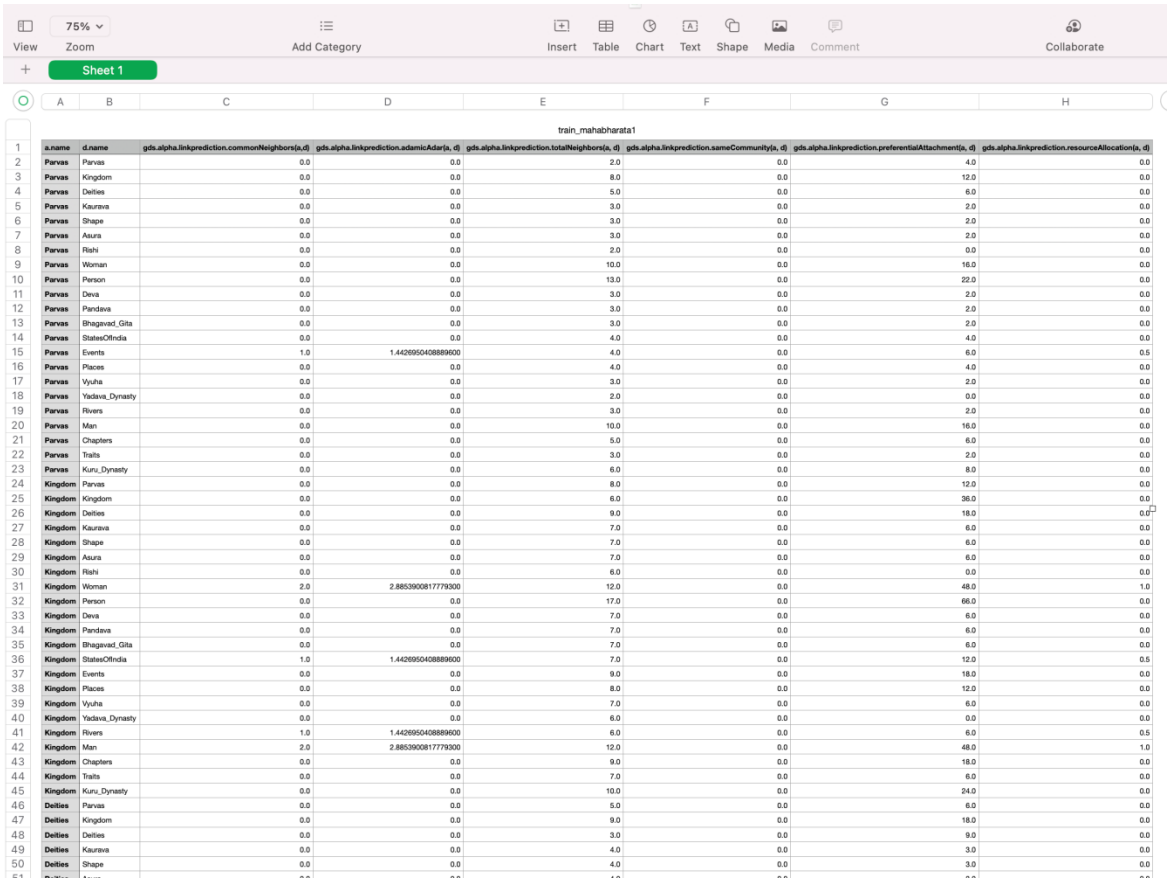
Adamic Adar:

It computes the sum of the inverse log of the degree of each of the neighbors. The degree of a node is the number of neighbors it has, and the intuition behind this algorithm is that when it comes to closing triangles, nodes of low degree are likely to be more influential.

Measure of Proximity for all classes



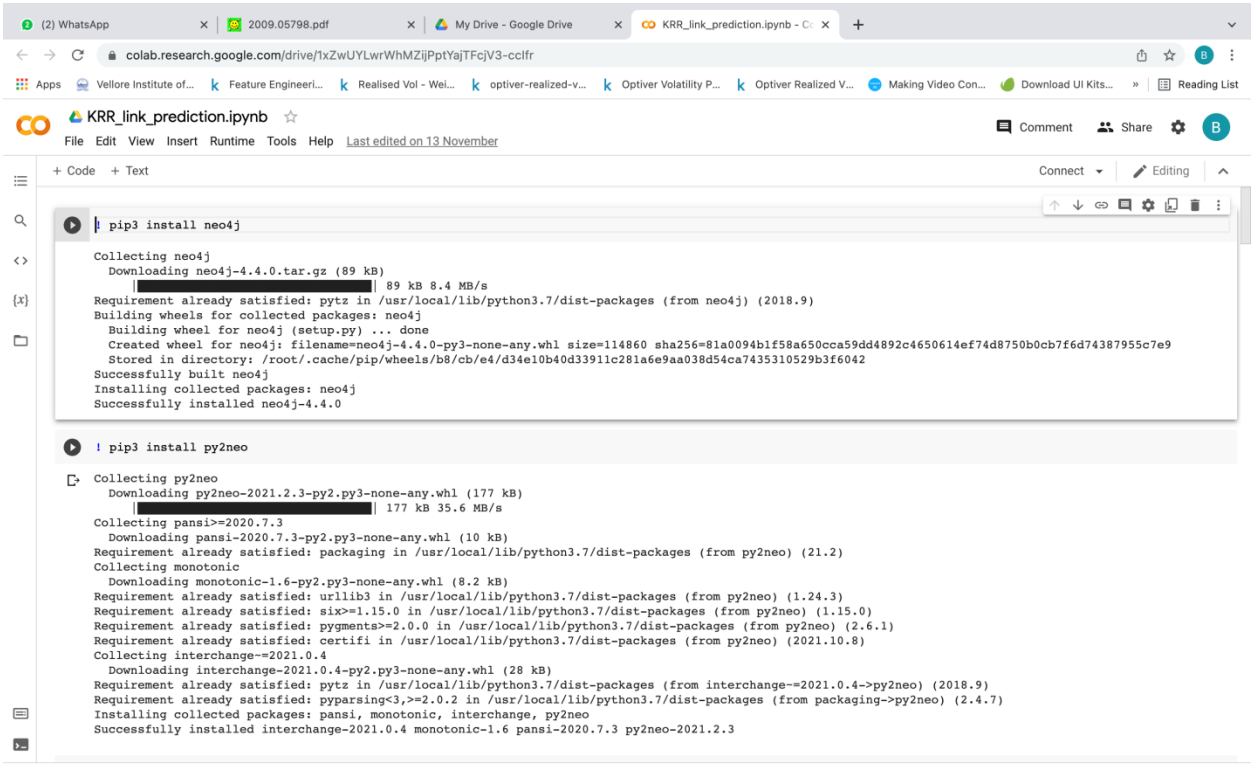
Export as CSV



train_mahabharata1

a.name	d.name	gds.alpha.linkprediction.commonNeighbors(a,d)	gds.alpha.linkprediction.adamicAdar(a, d)	gds.alpha.linkprediction.totalNeighbors(a, d)	gds.alpha.linkprediction.sameCommunity(a, d)	gds.alpha.linkprediction.preferentialAttachment(a, d)	gds.alpha.linkprediction.resourceAllocation(a, d)
Parvas	Parvas	0.0	0.0	2.0	0.0	4.0	0.0
Parvas	Kingdom	0.0	0.0	8.0	0.0	12.0	0.0
Parvas	Deities	0.0	0.0	5.0	0.0	6.0	0.0
Parvas	Kaurava	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Shape	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Asura	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Rishi	0.0	0.0	2.0	0.0	0.0	0.0
Parvas	Woman	0.0	0.0	10.0	0.0	16.0	0.0
Parvas	Person	0.0	0.0	13.0	0.0	22.0	0.0
Parvas	Deva	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Pandava	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Bhagavad_Gita	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	StatesOfIndia	0.0	0.0	4.0	0.0	4.0	0.0
Parvas	Events	1.0	1.442695040889600	4.0	0.0	6.0	0.5
Parvas	Places	0.0	0.0	4.0	0.0	4.0	0.0
Parvas	Vyuha	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Yadava_Dynasty	0.0	0.0	2.0	0.0	0.0	0.0
Parvas	Rivers	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Man	0.0	0.0	10.0	0.0	16.0	0.0
Parvas	Chapters	0.0	0.0	5.0	0.0	6.0	0.0
Parvas	Traits	0.0	0.0	3.0	0.0	2.0	0.0
Parvas	Kuru_Dynasty	0.0	0.0	6.0	0.0	8.0	0.0
Kingdom	Parvas	0.0	0.0	8.0	0.0	12.0	0.0
Kingdom	Kingdom	0.0	0.0	6.0	0.0	36.0	0.0
Kingdom	Deities	0.0	0.0	9.0	0.0	18.0	0.0
Kingdom	Kaurava	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Shape	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Asura	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Rishi	0.0	0.0	6.0	0.0	0.0	0.0
Kingdom	Woman	2.0	2.885390081779300	12.0	0.0	48.0	1.0
Kingdom	Person	0.0	0.0	17.0	0.0	66.0	0.0
Kingdom	Deva	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Pandava	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Bhagavad_Gita	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	StatesOfIndia	1.0	1.442695040889600	7.0	0.0	12.0	0.5
Kingdom	Events	0.0	0.0	9.0	0.0	18.0	0.0
Kingdom	Places	0.0	0.0	8.0	0.0	12.0	0.0
Kingdom	Vyuha	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Yadava_Dynasty	0.0	0.0	6.0	0.0	0.0	0.0
Kingdom	Rivers	1.0	1.442695040889600	6.0	0.0	6.0	0.5
Kingdom	Man	2.0	2.885390081779300	12.0	0.0	48.0	1.0
Kingdom	Chapters	0.0	0.0	9.0	0.0	18.0	0.0
Kingdom	Traits	0.0	0.0	7.0	0.0	6.0	0.0
Kingdom	Kuru_Dynasty	0.0	0.0	10.0	0.0	24.0	0.0
Deities	Parvas	0.0	0.0	5.0	0.0	6.0	0.0
Deities	Kingdom	0.0	0.0	9.0	0.0	18.0	0.0
Deities	Deities	0.0	0.0	3.0	0.0	9.0	0.0
Deities	Kaurava	0.0	0.0	4.0	0.0	3.0	0.0
Deities	Shape	0.0	0.0	4.0	0.0	3.0	0.0

Linking Neo4j with Google Colab

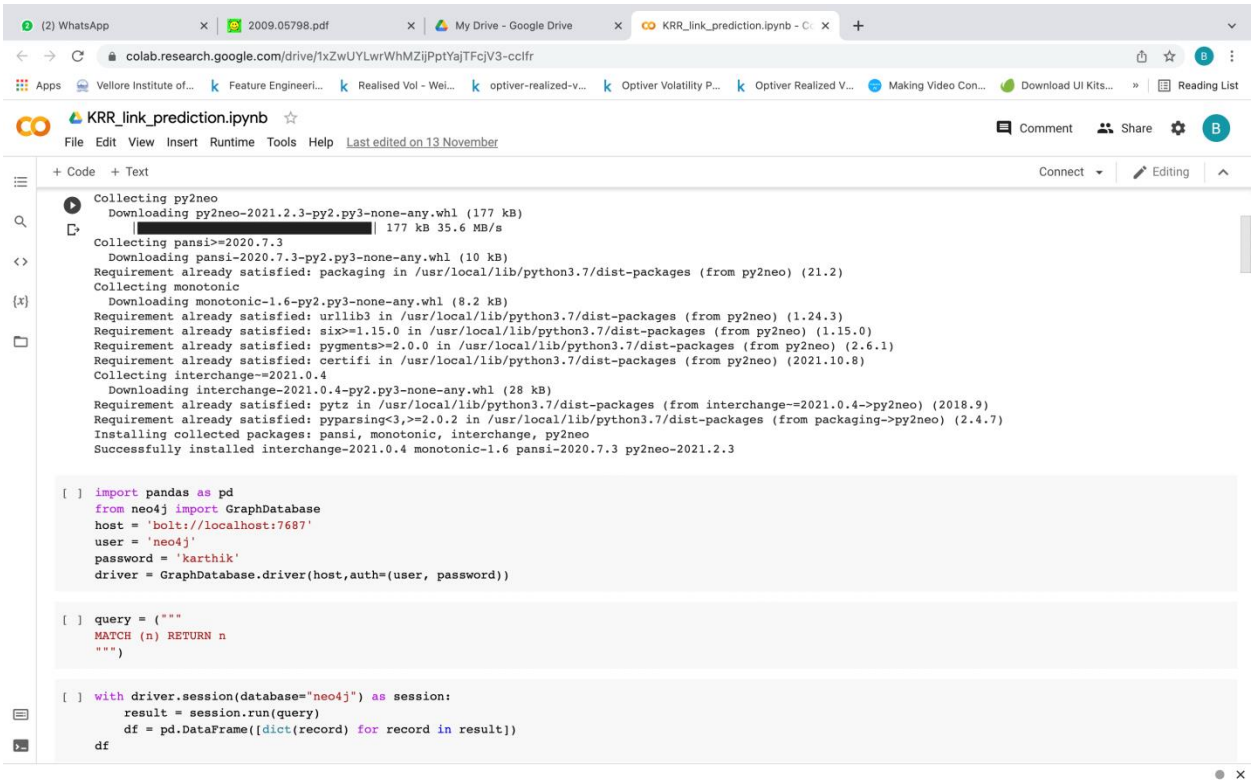


```
! pip3 install neo4j

Collecting neo4j
  Downloading neo4j-4.4.0.tar.gz (89 kB)
     |████████████████████████████████| 89 kB 8.4 MB/s
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from neo4j) (2018.9)
Building wheels for collected packages: neo4j
  Building wheel for neo4j (setup.py) ... done
  Created wheel for neo4j: filename=neo4j-4.4.0-py3-none-any.whl size=114860 sha256=81a0094b1f58a650cca59dd4892c4650614ef74d8750b0cb7f6d74387955c7e9
  Stored in directory: /root/.cache/pip/wheels/b8/cb/e4/d34e10b40d33911c281a6e9aa038d54ca7435310529b3f6042
Successfully built neo4j
Installing collected packages: neo4j
Successfully installed neo4j-4.4.0
```

```
! pip3 install py2neo

Collecting py2neo
  Downloading py2neo-2021.2.3-py2.py3-none-any.whl (177 kB)
     |████████████████████████████████| 177 kB 35.6 MB/s
Collecting pansi>=2020.7.3
  Downloading pansi-2020.7.3-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from py2neo) (21.2)
Collecting monotonic
  Downloading monotonic-1.6-py2.py3-none-any.whl (8.2 kB)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from py2neo) (1.24.3)
Requirement already satisfied: six>=1.15.0 in /usr/local/lib/python3.7/dist-packages (from py2neo) (1.15.0)
Requirement already satisfied: pygments>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from py2neo) (2.6.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from py2neo) (2021.10.8)
Collecting interchange~=2021.0.4
  Downloading interchange-2021.0.4-py2.py3-none-any.whl (28 kB)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from interchange~=2021.0.4->py2neo) (2018.9)
Requirement already satisfied: pyparsing<3,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->py2neo) (2.4.7)
Installing collected packages: pansi, monotonic, interchange, py2neo
Successfully installed interchange-2021.0.4 monotonic-1.6 pansi-2020.7.3 py2neo-2021.2.3
```



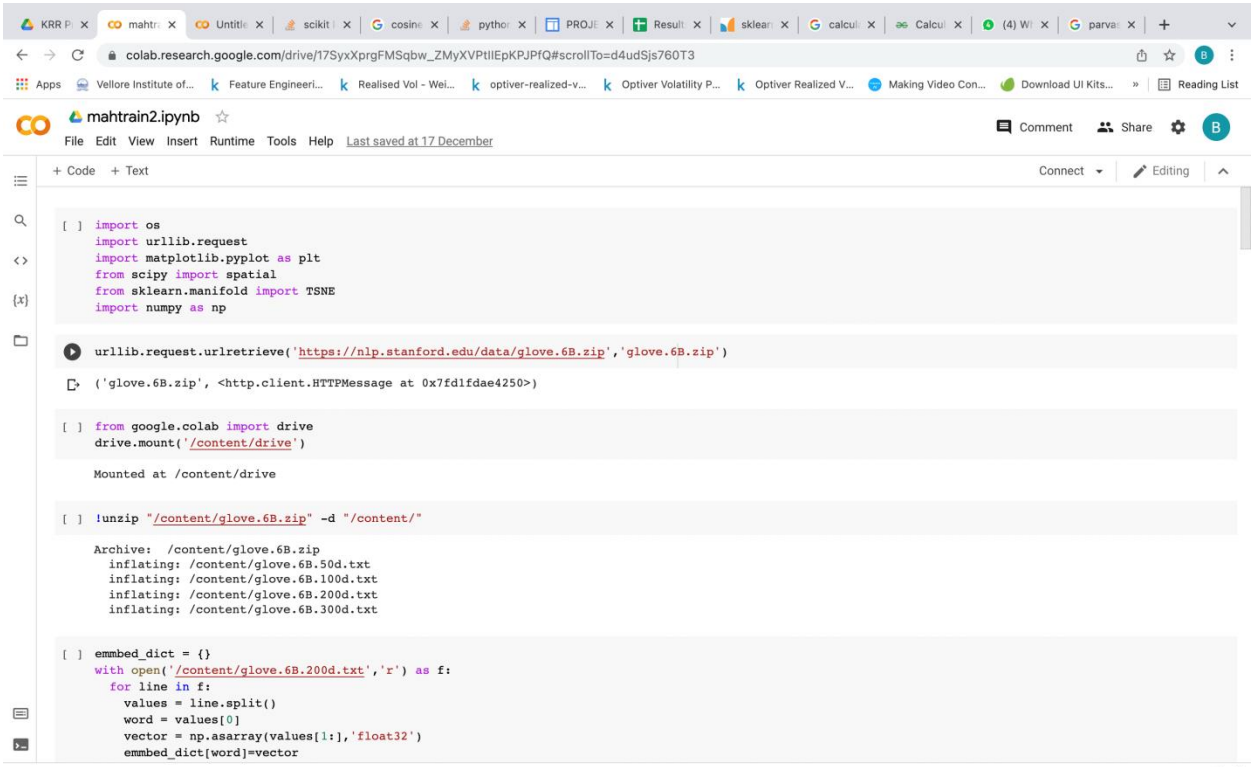
```
Collecting py2neo
  Downloading py2neo-2021.2.3-py2.py3-none-any.whl (177 kB)
     |████████████████████████████████| 177 kB 35.6 MB/s
Collecting pansi>=2020.7.3
  Downloading pansi-2020.7.3-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from py2neo) (21.2)
Collecting monotonic
  Downloading monotonic-1.6-py2.py3-none-any.whl (8.2 kB)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from py2neo) (1.24.3)
Requirement already satisfied: six>=1.15.0 in /usr/local/lib/python3.7/dist-packages (from py2neo) (1.15.0)
Requirement already satisfied: pygments>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from py2neo) (2.6.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from py2neo) (2021.10.8)
Collecting interchange~=2021.0.4
  Downloading interchange-2021.0.4-py2.py3-none-any.whl (28 kB)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from interchange~=2021.0.4->py2neo) (2018.9)
Requirement already satisfied: pyparsing<3,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->py2neo) (2.4.7)
Installing collected packages: pansi, monotonic, interchange, py2neo
Successfully installed interchange-2021.0.4 monotonic-1.6 pansi-2020.7.3 py2neo-2021.2.3
```

```
[ ] import pandas as pd
    from neo4j import GraphDatabase
    host = 'bolt://localhost:7687'
    user = 'neo4j'
    password = 'karthik'
    driver = GraphDatabase.driver(host,auth=(user, password))
```

```
[ ] query = ("""
    MATCH (n) RETURN n
    """)
```

```
[ ] with driver.session(database="neo4j") as session:
        result = session.run(query)
        df = pd.DataFrame([dict(record) for record in result])
    df
```

GloVe Embedding



mahtrain2.ipynb

File Edit View Insert Runtime Tools Help Last saved at 17 December

Comment Share

+ Code + Text

Connect Editing

```
[ ] import os
    import urllib.request
    import matplotlib.pyplot as plt
    from scipy import spatial
    from sklearn.manifold import TSNE
    import numpy as np
```

```
urllib.request.urlretrieve('https://nlp.stanford.edu/data/glove.6B.zip','glove.6B.zip')
```

```
('glove.6B.zip', <http.client.HTTPMessage at 0x7fd1fdae4250>)
```

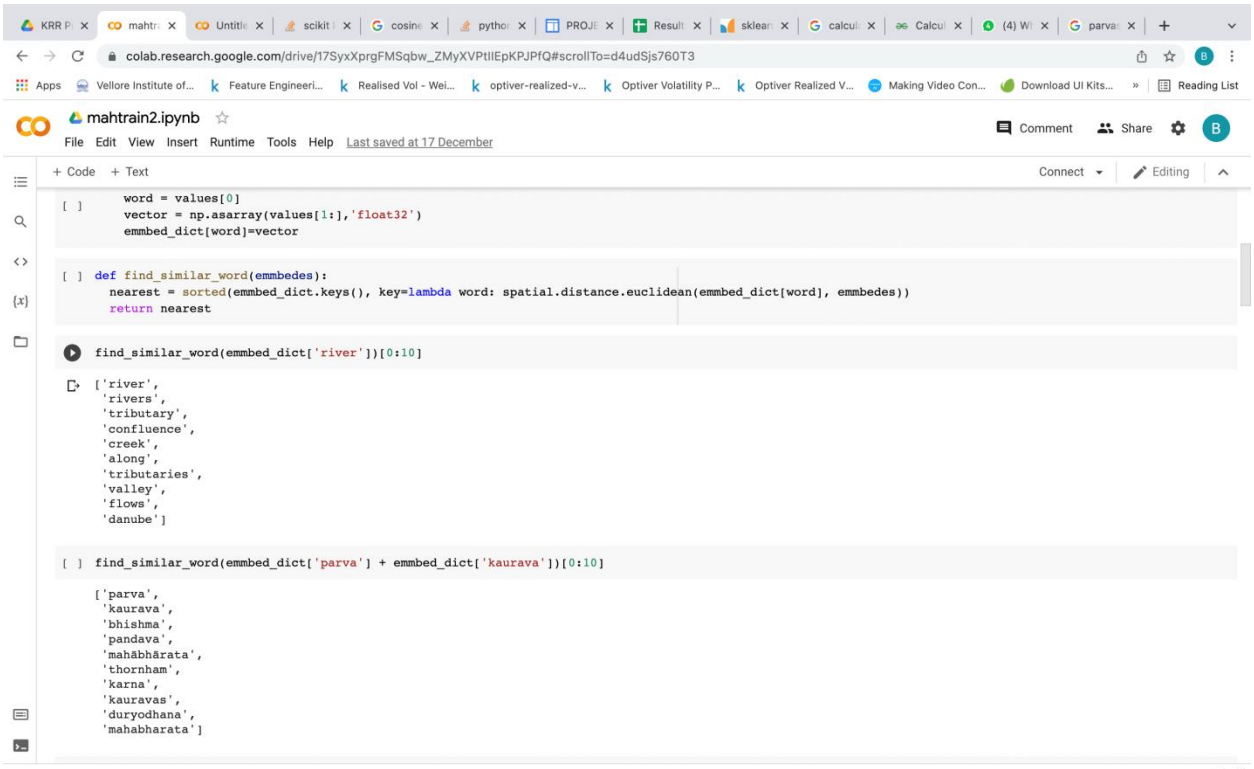
```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] !unzip "/content/glove.6B.zip" -d "/content/"
```

```
Archive:  /content/glove.6B.zip
  inflating: /content/glove.6B.50d.txt
  inflating: /content/glove.6B.100d.txt
  inflating: /content/glove.6B.200d.txt
  inflating: /content/glove.6B.300d.txt
```

```
[ ] emmbed_dict = {}
    with open('/content/glove.6B.200d.txt','r') as f:
      for line in f:
        values = line.split()
        word = values[0]
        vector = np.asarray(values[1:],'float32')
        emmbed_dict[word]=vector
```



mahtrain2.ipynb

File Edit View Insert Runtime Tools Help Last saved at 17 December

Comment Share

+ Code + Text

Connect Editing

```
[ ]     word = values[0]
        vector = np.asarray(values[1:],'float32')
        emmbed_dict[word]=vector
```

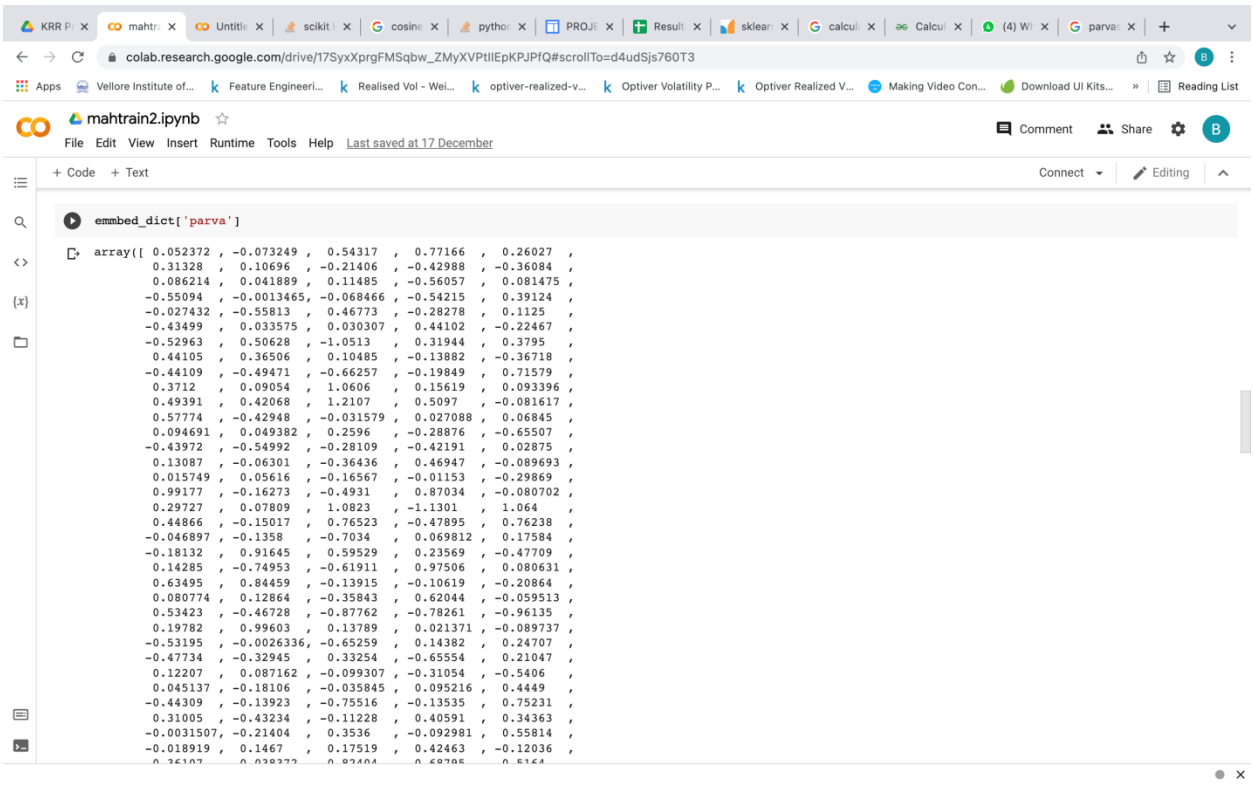
```
[ ] def find_similar_word(emmbedes):
      nearest = sorted(emmbed_dict.keys(), key=lambda word: spatial.distance.euclidean(emmbed_dict[word], emmbedes))
      return nearest
```

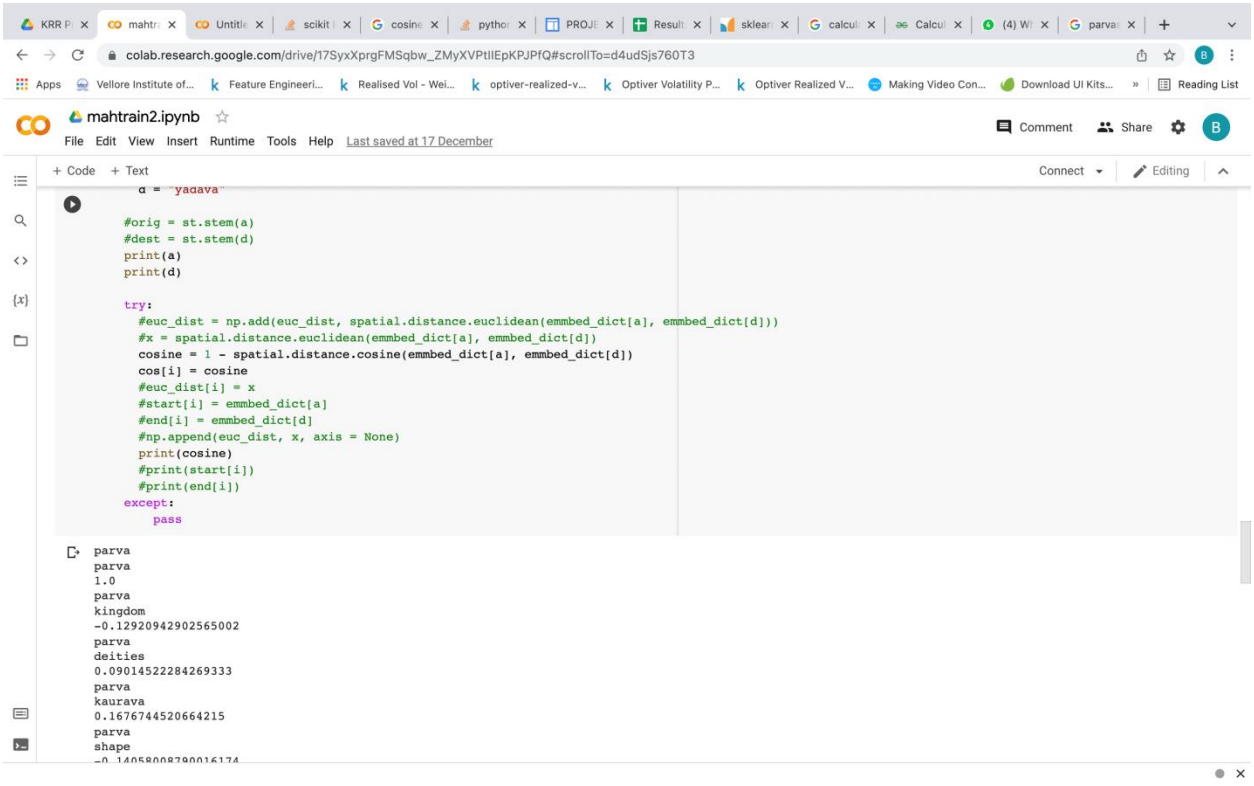
```
find_similar_word(emmbed_dict['river'])[0:10]
```

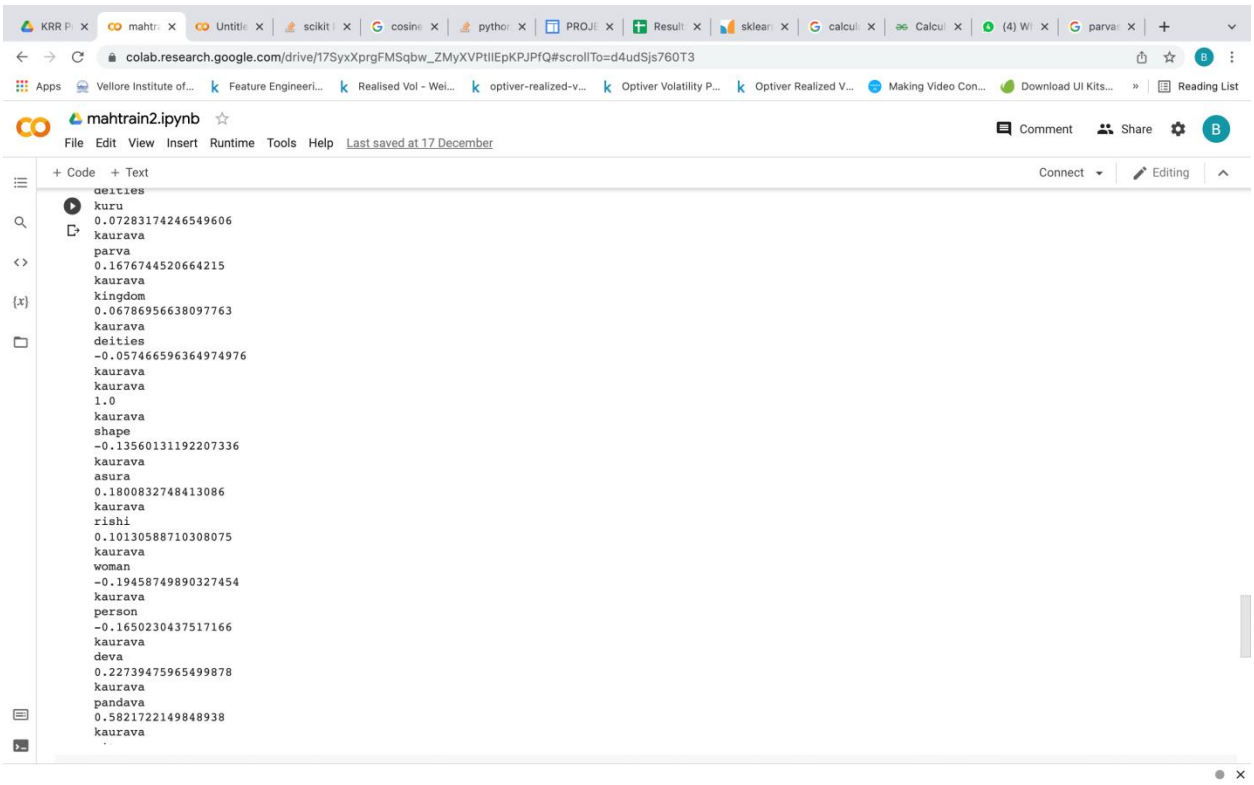
```
['river',
 'rivers',
 'tributary',
 'confluence',
 'creek',
 'along',
 'tributaries',
 'valley',
 'flows',
 'danube']
```

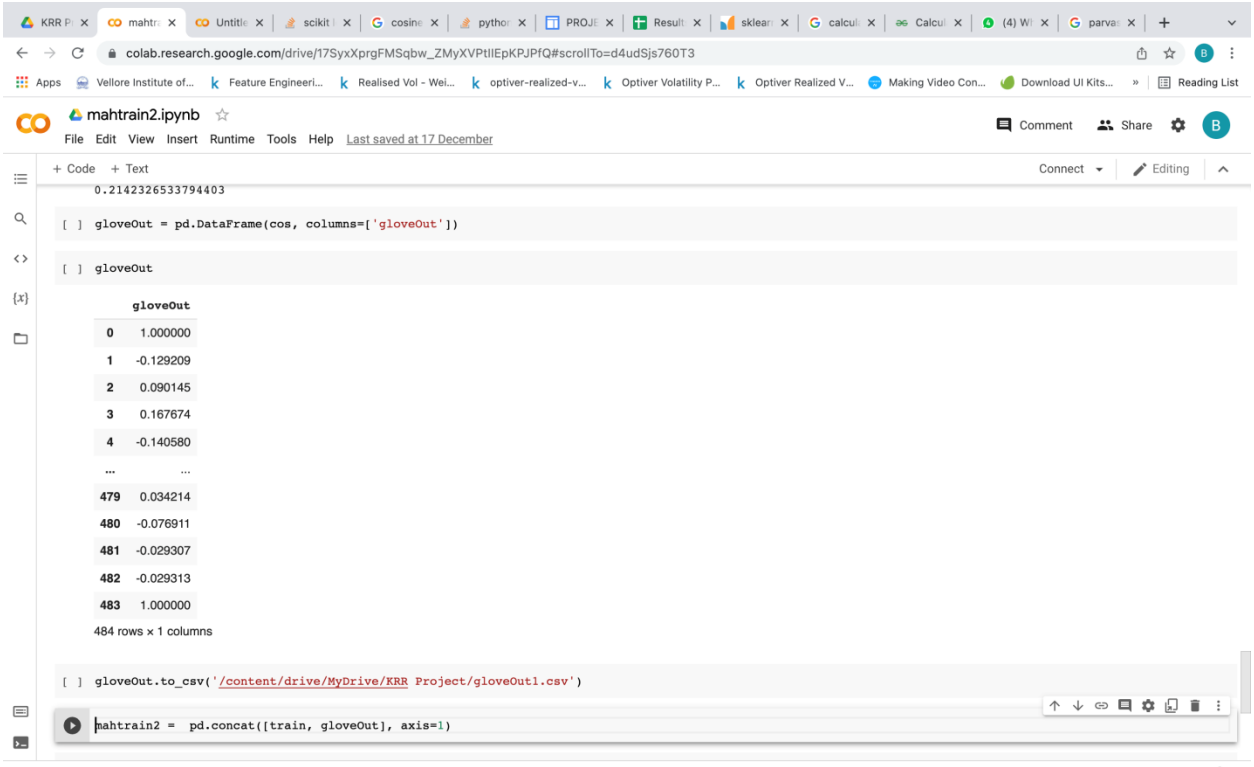
```
[ ] find_similar_word(emmbed_dict['parva'] + emmbed_dict['kaurava'])[0:10]
```

```
['parva',
 'kaurava',
 'bhishma',
 'pandava',
 'mahābhārata',
 'thornham',
 'karna',
 'kauravas',
 'duryodhana',
 'mahabharata']
```

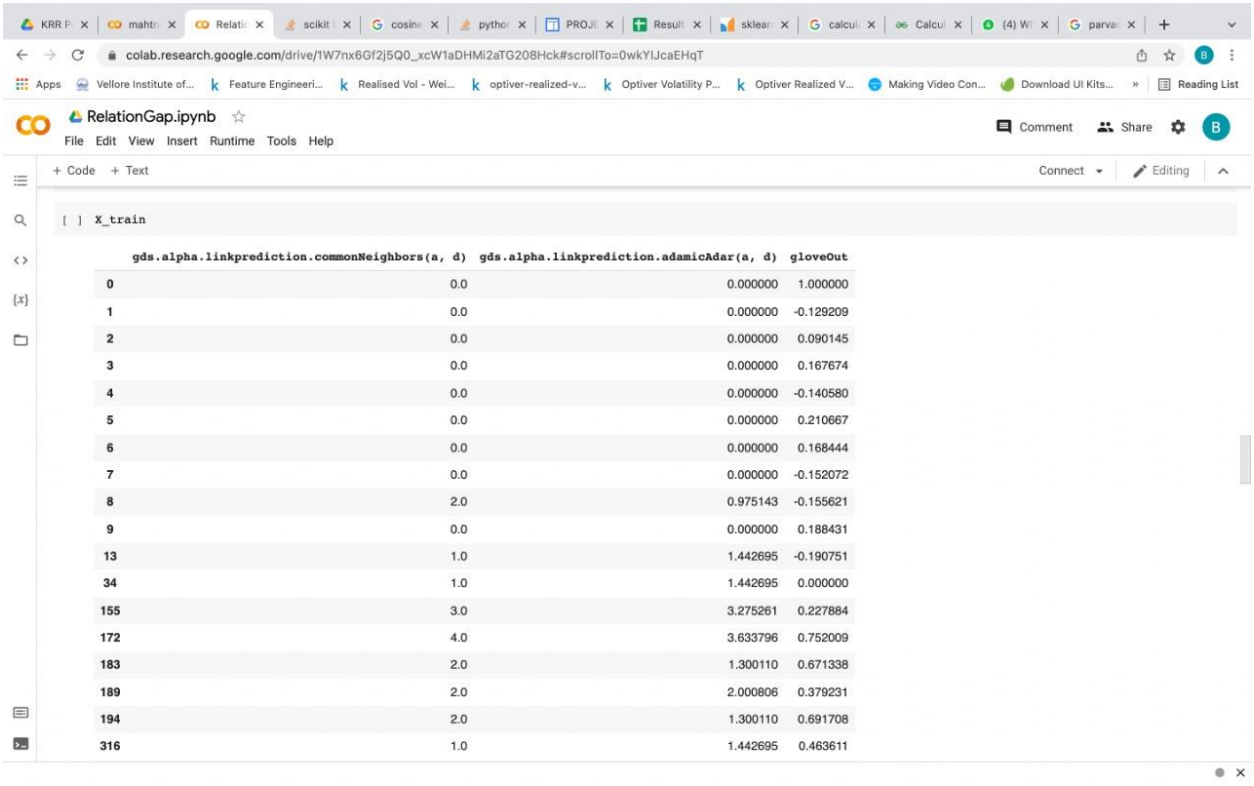

colab.research.google.com/drive/17SyxXprgFMSqbw_ZMyXVPtIlEpKPJPfQ#scrollTo=d4udSjs760T3
mahtrain2.ipynb
File Edit View Insert Runtime Tools Help Last saved at 17 December
Comment
Share
+ Code + Text
Connect
Editing
emmbed_dict['parva']
array([0.052372 , -0.073249 , 0.54317 , 0.77166 , 0.26027 ,
0.31328 , 0.10696 , -0.21406 , -0.42988 , -0.36084 ,
0.086214 , 0.041889 , 0.11485 , -0.56057 , 0.081475 ,
-0.55094 , -0.0013465, -0.068466 , -0.54215 , 0.39124 ,
-0.027432 , -0.55813 , 0.46773 , -0.28278 , 0.1125   ,
-0.43499 , 0.033575 , 0.030307 , 0.44102 , -0.22467 ,
-0.52963 , 0.50628 , -1.0513   , 0.31944 , 0.3795   ,
0.44105 , 0.36506 , 0.10485 , -0.13882 , -0.36718 ,
-0.44109 , -0.49471 , -0.66257 , -0.19849 , 0.71579 ,
0.3712   , 0.09054 , 1.0606   , 0.15619 , 0.093396 ,
0.49391 , 0.42068 , 1.2107   , 0.5097   , -0.081617 ,
0.57774 , -0.42948 , -0.031579 , 0.027088 , 0.06845 ,
0.094691 , 0.049382 , 0.2596   , -0.28876 , -0.65507 ,
-0.43972 , -0.54992 , -0.28109 , -0.42191 , 0.02875 ,
0.13087 , -0.06301 , -0.36436 , 0.46947 , -0.089693 ,
0.015749 , 0.05616 , -0.16567 , -0.01153 , -0.29869 ,
0.99177 , -0.16273 , -0.4931   , 0.87034 , -0.080702 ,
0.29727 , 0.07809 , 1.0823   , -1.1301   , 1.064   ,
0.44866 , -0.15017 , 0.76523 , -0.47895 , 0.76238 ,
-0.046897 , -0.1358   , -0.7034   , 0.069812 , 0.17584 ,
-0.18132 , 0.91645 , 0.59529 , 0.23569 , -0.47709 ,
0.14285 , -0.74953 , -0.61911 , 0.97506 , 0.080631 ,
0.63495 , 0.84459 , -0.13915 , -0.10619 , -0.20864 ,
0.080774 , 0.12864 , -0.35843 , 0.62044 , -0.059513 ,
0.53423 , -0.46728 , -0.87762 , -0.78261 , -0.96135 ,
0.19782 , 0.99603 , 0.13789 , 0.021371 , -0.089737 ,
-0.53195 , -0.0026336, -0.65259 , 0.14382 , 0.24707 ,
-0.47734 , -0.32945 , 0.33254 , -0.65554 , 0.21047 ,
0.12207 , 0.087162 , -0.099307 , -0.31054 , -0.5406   ,
0.045137 , -0.18106 , -0.035845 , 0.095216 , 0.4449   ,
-0.44309 , -0.13923 , -0.75516 , -0.13535 , 0.75231 ,
0.31005 , -0.43234 , -0.11228 , 0.40591 , 0.34363 ,
-0.0031507, -0.21404 , 0.3536   , -0.092981 , 0.55814 ,
-0.018919 , 0.1467   , 0.17519 , 0.42463 , -0.12036 ,


colab.research.google.com/drive/17SyxXprgFMSqbw_ZMyXVPtIlEpKPJPfQ#scrollTo=d4udSjs760T3
mahtrain2.ipynb
File Edit View Insert Runtime Tools Help Last saved at 17 December
Comment
Share
+ Code + Text
Connect
Editing
a = "yadava"
#orig = st.stem(a)
#dest = st.stem(d)
print(a)
print(d)
try:
#euc_dist = np.add(euc_dist, spatial.distance.euclidean(emmbed_dict[a], emmbed_dict[d]))
#x = spatial.distance.euclidean(emmbed_dict[a], emmbed_dict[d])
cosine = 1 - spatial.distance.cosine(emmbed_dict[a], emmbed_dict[d])
cos[i] = cosine
#euc_dist[i] = x
#start[i] = emmbed_dict[a]
#end[i] = emmbed_dict[d]
#np.append(euc_dist, x, axis = None)
print(cosine)
#print(start[i])
#print(end[i])
except:
pass
parva
parva
1.0
parva
kingdom
-0.12920942902565002
parva
deities
0.09014522284269333
parva
kaurava
0.1676744520664215
parva
shape


colab.research.google.com/drive/17SyxXprgFMSqbw_ZMyXVPtIlEpKPJPfQ#scrollTo=d4udSjs760T3
mahtrain2.ipynb
File Edit View Insert Runtime Tools Help Last saved at 17 December
Comment
Share
+ Code + Text
Connect
Editing
deities
kuru
0.07283174246549606
kaurava
parva
0.1676744520664215
kaurava
kingdom
0.06786956638097763
kaurava
deities
-0.057466596364974976
kaurava
kaurava
1.0
kaurava
shape
-0.13560131192207336
kaurava
asura
0.1800832748413086
kaurava
rishi
0.10130588710308075
kaurava
woman
-0.19458749890327454
kaurava
person
-0.1650230437517166
kaurava
deva
0.22739475965499878
kaurava
pandava
0.5821722149848938
kaurava


colab.research.google.com/drive/17SyxXprgFMSqbw_ZMyXVPtIlEpKPJPfQ#scrollTo=d4udSjs760T3
mahtrain2.ipynb
File Edit View Insert Runtime Tools Help Last saved at 17 December
Comment
Share
+ Code + Text
Connect
Editing
0.2142326533794403
[ ] gloveOut = pd.DataFrame(cos, columns=['gloveOut'])
[ ] gloveOut
gloveOut
0 1.000000
1 -0.129209
2 0.090145
3 0.167674
4 -0.140580
... ...
479 0.034214
480 -0.076911
481 -0.029307
482 -0.029313
483 1.000000
484 rows × 1 columns
[ ] gloveOut.to_csv('/content/drive/MyDrive/KRR Project/gloveOut1.csv')
mahtrain2 = pd.concat([train, gloveOut], axis=1)

SVM Model



RelationGap.ipynb

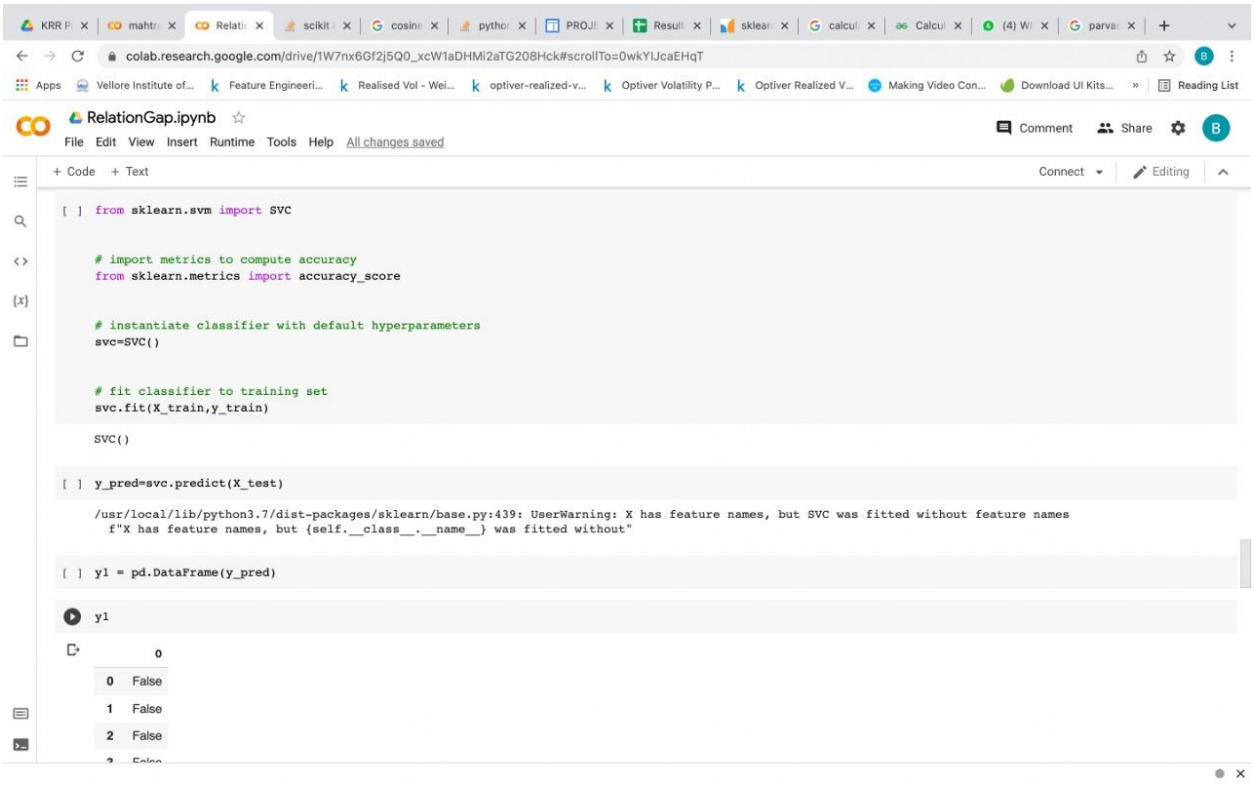
File Edit View Insert Runtime Tools Help

+ Code + Text

Connect Editing

```
[ ] X_train
```

	gds.alpha.linkprediction.commonNeighbors(a, d)	gds.alpha.linkprediction.adamicAdar(a, d)	gloveOut
0	0.0	0.000000	1.000000
1	0.0	0.000000	-0.129209
2	0.0	0.000000	0.090145
3	0.0	0.000000	0.167674
4	0.0	0.000000	-0.140580
5	0.0	0.000000	0.210667
6	0.0	0.000000	0.168444
7	0.0	0.000000	-0.152072
8	2.0	0.975143	-0.155621
9	0.0	0.000000	0.188431
13	1.0	1.442695	-0.190751
34	1.0	1.442695	0.000000
155	3.0	3.275261	0.227884
172	4.0	3.633796	0.752009
183	2.0	1.300110	0.671338
189	2.0	2.000806	0.379231
194	2.0	1.300110	0.691708
316	1.0	1.442695	0.463611



RelationGap.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Connect Editing

```
[ ] from sklearn.svm import SVC


    # import metrics to compute accuracy
    from sklearn.metrics import accuracy_score


    # instantiate classifier with default hyperparameters
    svc=SVC()


    # fit classifier to training set
    svc.fit(X_train,y_train)
```

```
SVC()
```

```
[ ] y_pred=svc.predict(X_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:439: UserWarning: X has feature names, but SVC was fitted without feature names
  f"X has feature names, but {self.__class__.__name__} was fitted without"
```

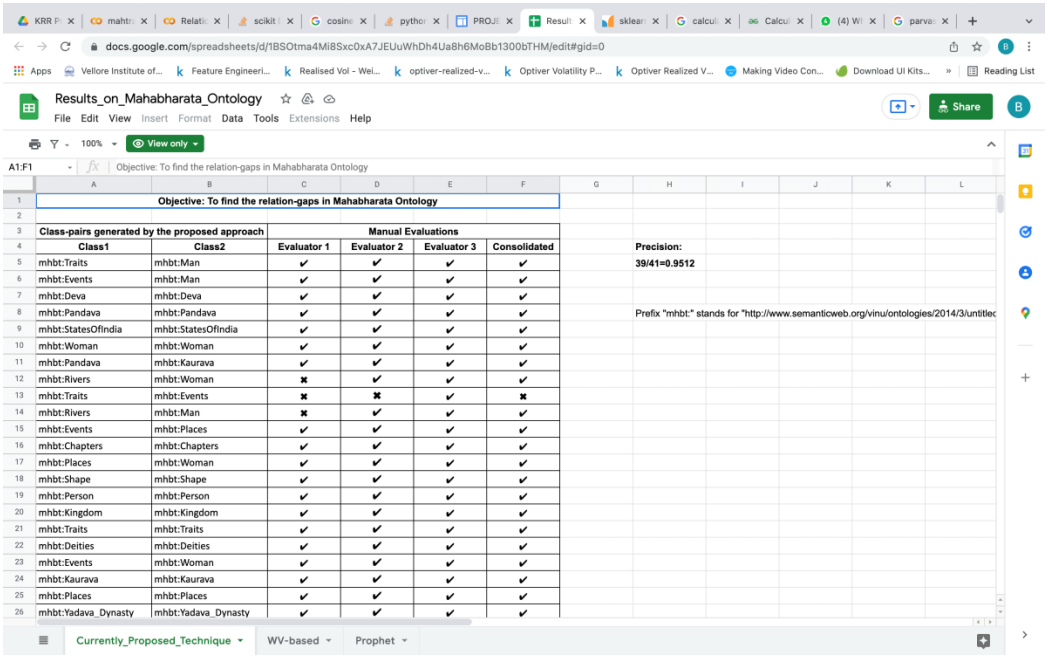
```
[ ] y1 = pd.DataFrame(y_pred)
```

```
y1
```

	0
0	False
1	False
2	False

Results

The output we should get



Results_on_Mahabharata_Ontology

Objective: To find the relation-gaps in Mahabharata Ontology

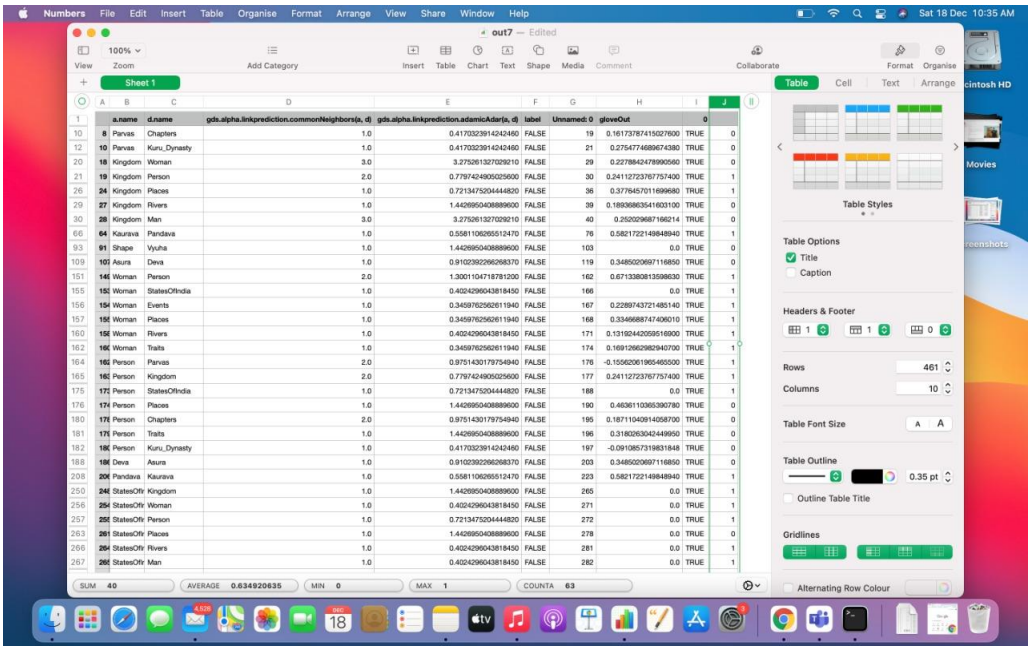
Class-pairs generated by the proposed approach		Manual Evaluations			
Class1	Class2	Evaluator 1	Evaluator 2	Evaluator 3	Consolidated
mhbt:Traits	mhbt:Man	✔	✔	✔	✔
mhbt:Events	mhbt:Man	✔	✔	✔	✔
mhbt:Deva	mhbt:Deva	✔	✔	✔	✔
mhbt:Pandava	mhbt:Pandava	✔	✔	✔	✔
mhbt:StatesOfIndia	mhbt:StatesOfIndia	✔	✔	✔	✔
mhbt:Woman	mhbt:Woman	✔	✔	✔	✔
mhbt:Pandava	mhbt:Kaurava	✔	✔	✔	✔
mhbt:Rivers	mhbt:Woman	✖	✔	✔	✔
mhbt:Traits	mhbt:Events	✖	✖	✔	✖
mhbt:Rivers	mhbt:Man	✖	✔	✔	✔
mhbt:Events	mhbt:Places	✔	✔	✔	✔
mhbt:Chapters	mhbt:Chapters	✔	✔	✔	✔
mhbt:Places	mhbt:Woman	✔	✔	✔	✔
mhbt:Shape	mhbt:Shape	✔	✔	✔	✔
mhbt:Person	mhbt:Person	✔	✔	✔	✔
mhbt:Kingdom	mhbt:Kingdom	✔	✔	✔	✔
mhbt:Traits	mhbt:Traits	✔	✔	✔	✔
mhbt:Deities	mhbt:Deities	✔	✔	✔	✔
mhbt:Events	mhbt:Woman	✔	✔	✔	✔
mhbt:Kaurava	mhbt:Kaurava	✔	✔	✔	✔
mhbt:Places	mhbt:Places	✔	✔	✔	✔
mhbt:Yadava_Dynasty	mhbt:Yadava_Dynasty	✔	✔	✔	✔

Precision:
39/41=0.9512

Prefix "mhbt:" stands for "http://www.semanticweb.org/vinu/ontologies/2014/3/untitled

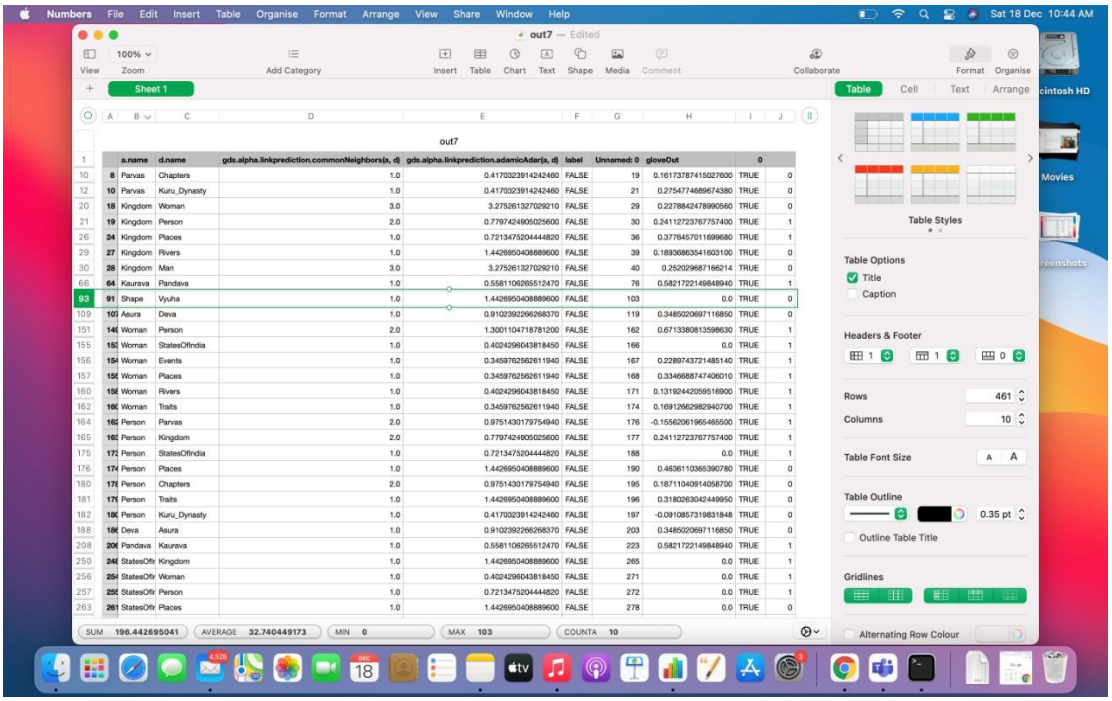
Currently_Proposed_Technique | WV-based | Prophet

Our Output

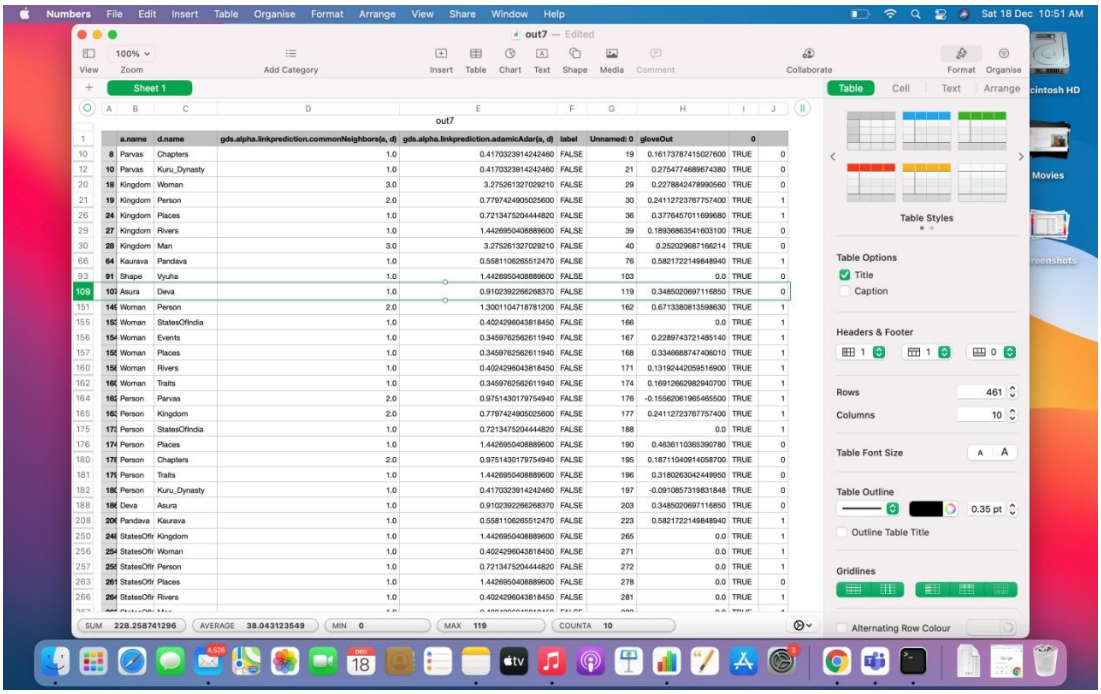


Totally they predicted 20 classes that could be connected in the given paper. We got all 20 of those outputs. In extra to that, we also got these nodes that could be connected in the following pictures.

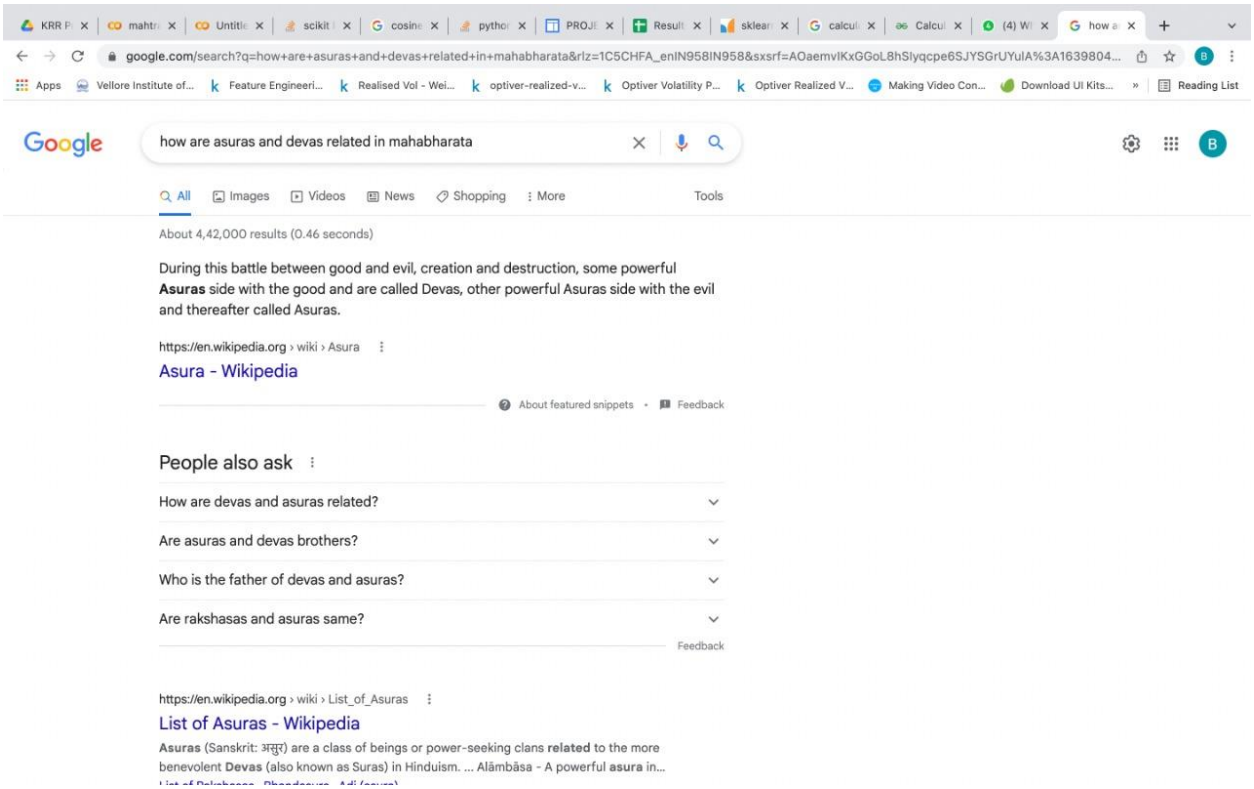
Shape - Vyuha



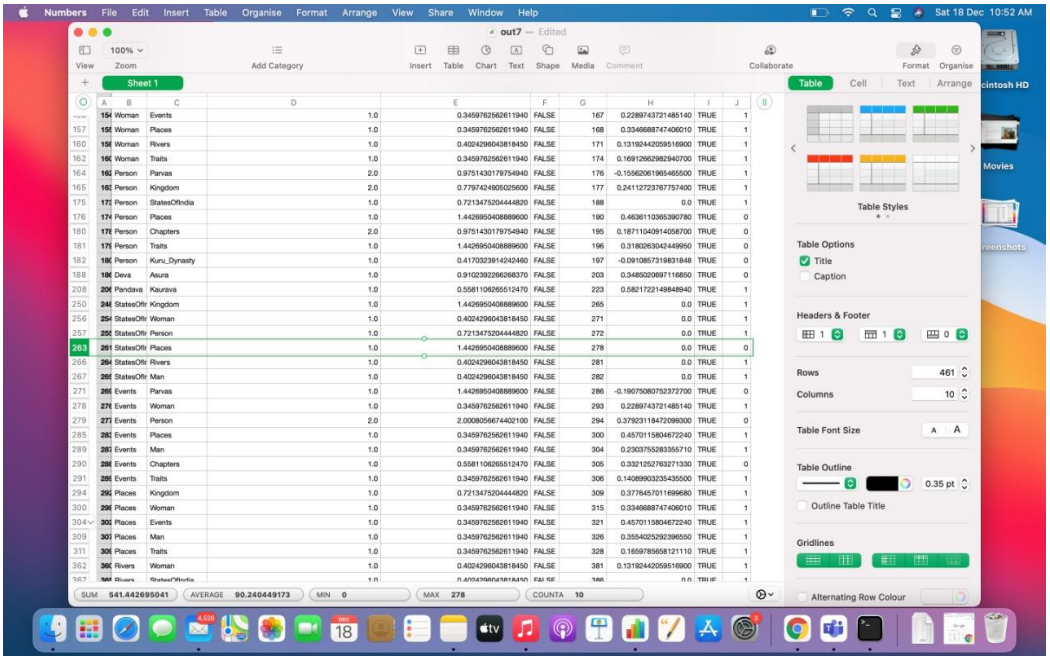
Asura – Deva



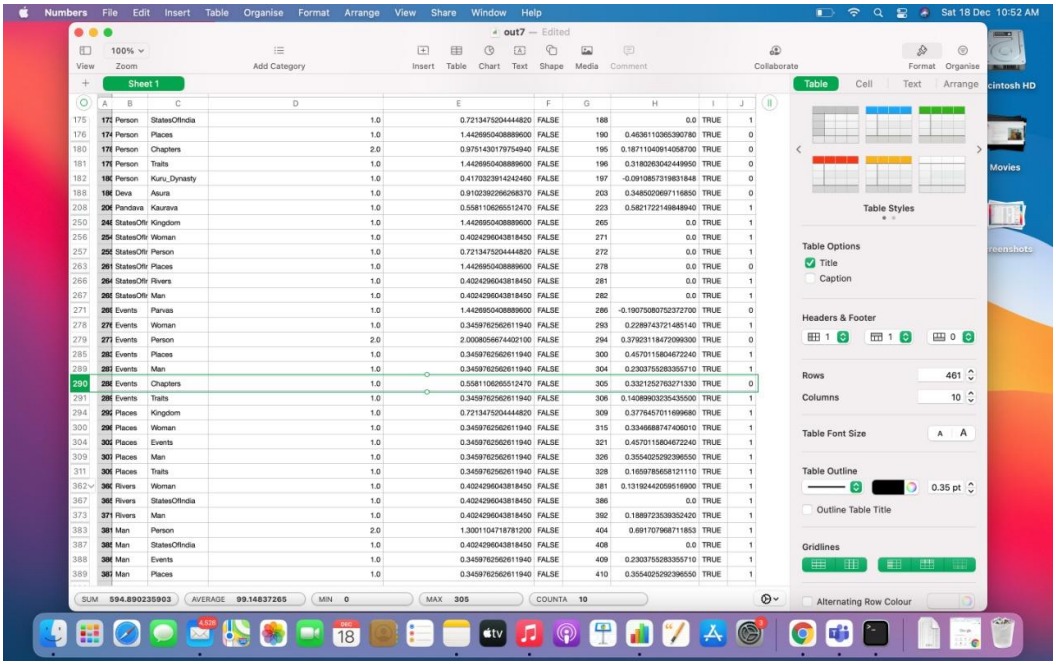
Proof of relationship between Asura and Deva:



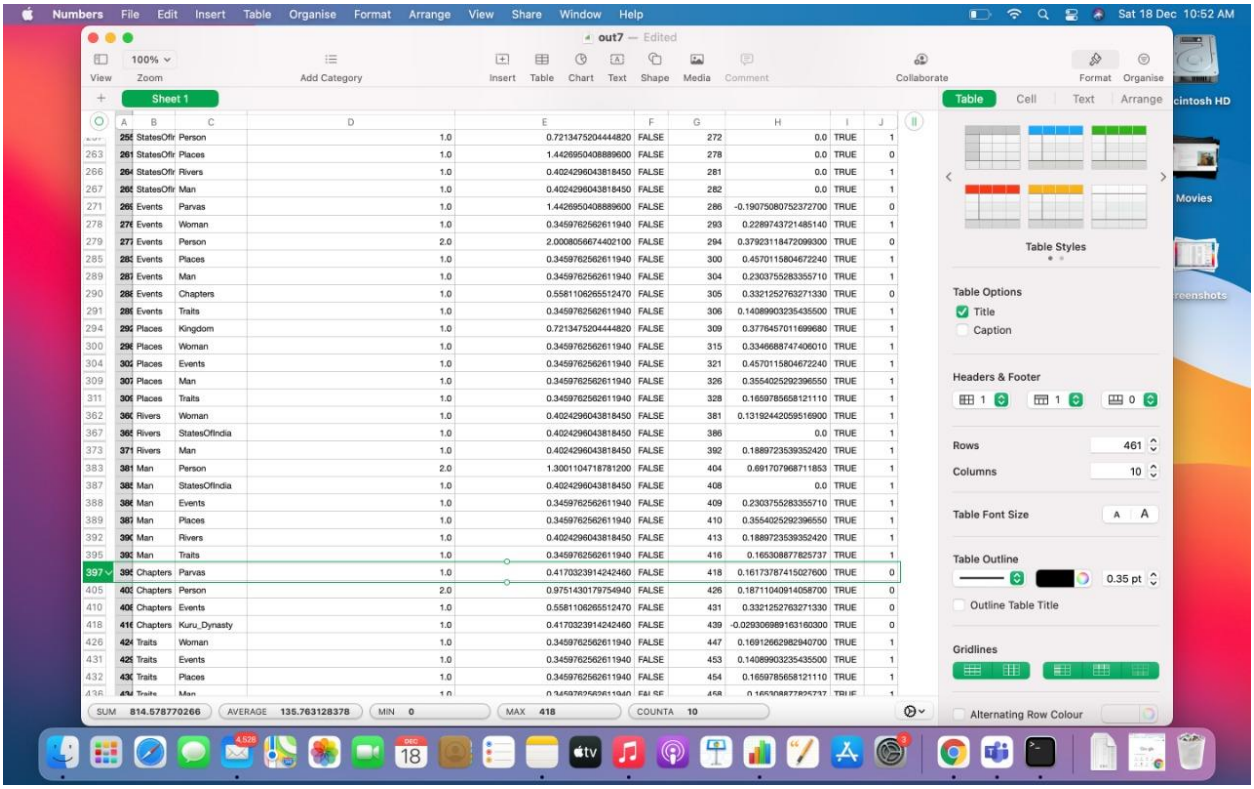
StatesOfIndia- Places



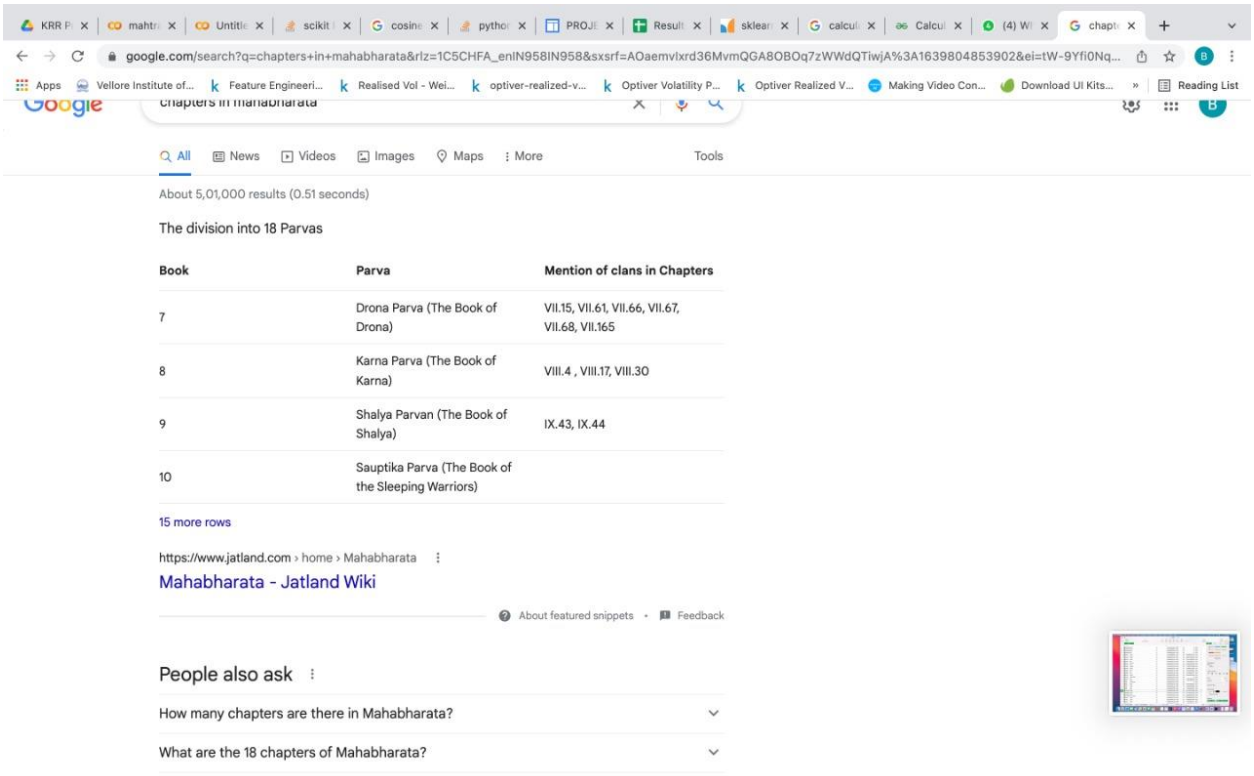
Events -Chapters



Chapters - Parvas



Proof of relationship between Chapters and Parvas:



About 5,01,000 results (0.51 seconds)

The division into 18 Parvas

Book	Parva	Mention of clans in Chapters
7	Drona Parva (The Book of Drona)	VII.15, VII.61, VII.66, VII.67, VII.68, VII.165
8	Karna Parva (The Book of Karna)	VIII.4 , VIII.17, VIII.30
9	Shalya Parvan (The Book of Shalya)	IX.43, IX.44
10	Sauptika Parva (The Book of the Sleeping Warriors)	

15 more rows

https://www.jatland.com › home › Mahabharata

Mahabharata - Jatland Wiki

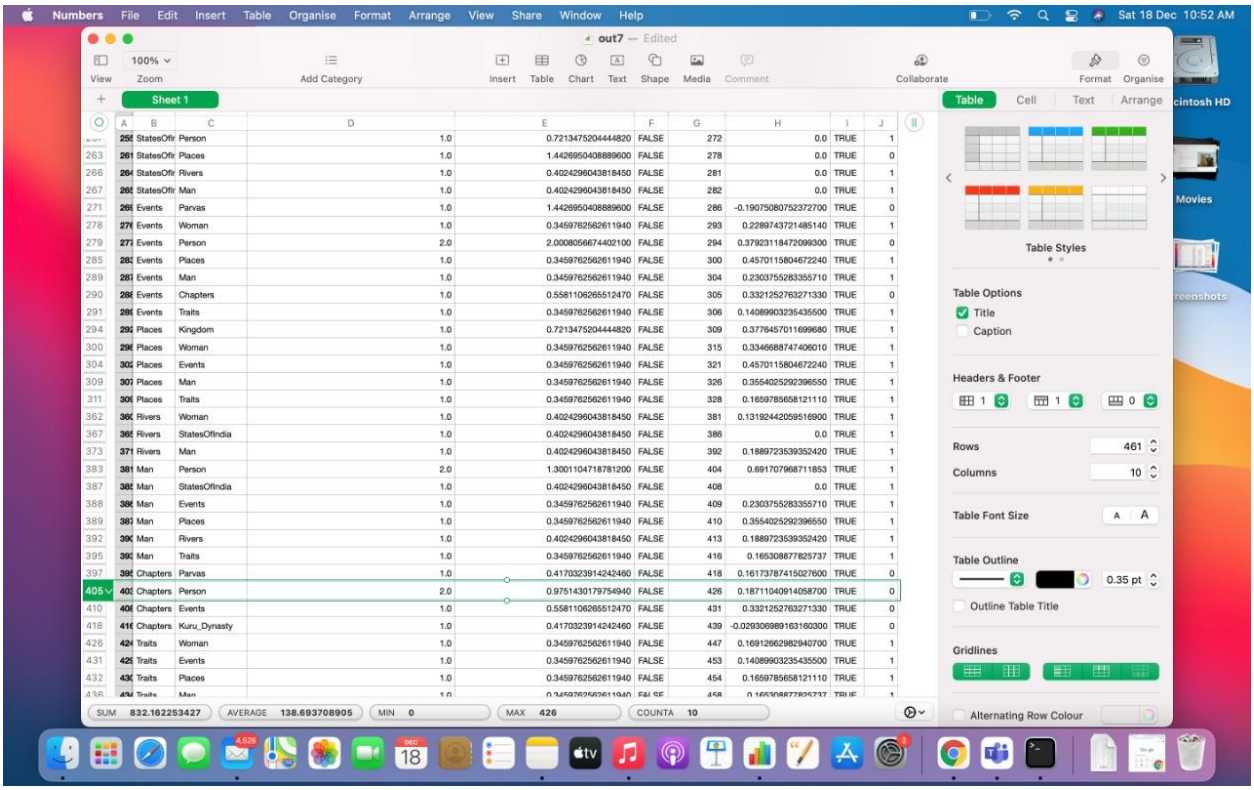
About featured snippets · Feedback

People also ask

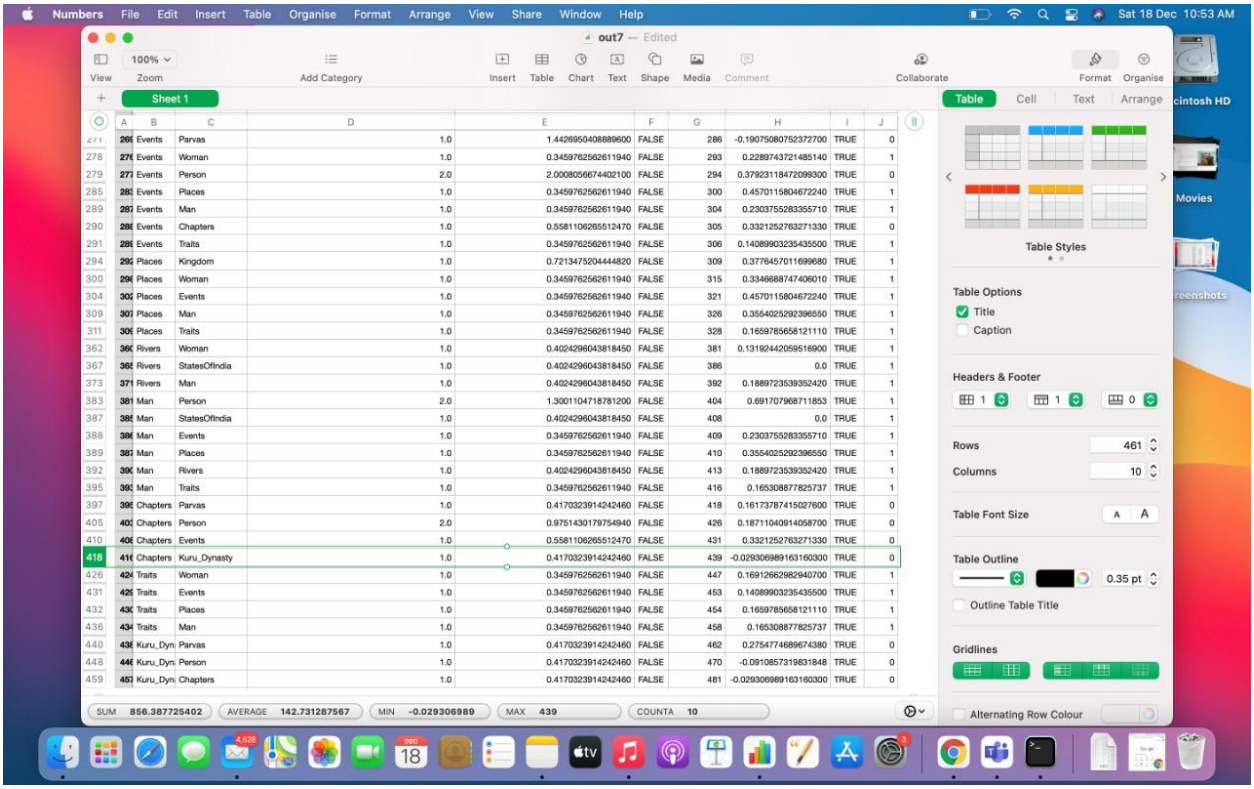
How many chapters are there in Mahabharata?

What are the 18 chapters of Mahabharata?

Chapters - Person



Chapters – Kuru_Dynasty



Total Relation Gaps predicted by our model: 36

Total Relation Gaps predicted by the model given in paper: 20

Extra relations we predicted: 16

Precision: 34/36 = **94.4%**