Zafir Khalid
40152164
Mini Project 2 – Report
COMP 472 NN 2214
Dr. Sabine Bergler

(a) SpaCy sentence and token splits for S1

Sentence splitting is done in main.py by using the following snippet

```
for sent in doc.sents:
```

Token splitting is done in main.py by using the following snippet

```
for sent in doc.sents:
        for token in sent:
```

Where doc is the text file loaded into spacy.

(b) a graphical representation of the two dependency graphs for S1

All graphs are saved as .html files under the Dependency Graphs folder.

(c) $T1_{S1}$ and $T2_{S1}$ for the snippet

$T1_{S1}$ and $T2_{S2}$ both created and handled using pandas in main.py
Note: Only a snippet of the table is shown in this report

$T1_{S1}$ – before processing (First 5 entries)

Token	NE?	NEtype	Governor	SentimentValueOfToken	SentimentValueOfSentence
U.S.	384	GPE	agencies	0.0	2.0
intelligence	0		agencies	0.0	2.0
agencies	0		concluded	0.0	2.0
concluded	0		concluded	0.0	2.0
in	0		concluded	0.0	2.0

$T2_{S1}$ – before processing (First 5 entries)

Token	NEtype	Governor	SentimentValueOfToken	SentimentValueOfSentence
U.S.	GPE	agencies	0.0	2.0
January 2017	DATE	in	0.0	2.0
Russia	GPE	mounted	0.0	2.0
Trump	PERSON	beat	0.0	2.0
Clinton	PERSON	beat	0.0	2.0

Note:

For clustering of APonTrump the same structure is used to create the tables but all text entries are encoded to numerical values. All strings from the text document are replaced by their vector norms provided by Spacy. All NEtype entrires are encoded using a label encoder provided by sklearn.

(d) 3-means clusters for T1 (for the entire text)

Clustering of T1:

Cluster 1:
['officer', 'venue', 'interference', 'lost', '34', 'lawyers', 'fixer', 'reputation', 'skyscraper', 'ranging', 'fixer', 'not', 'fact', 'settled', 'contrived', 'Times', 'hundreds', 'fortunes', 'critic', 'not']

Cluster 2:
['Lock', 'two', 'Gillers', 'credibility', 'accused', 'state', 'allies', '2020', 'suits', 'AP', 'Ted', 'election', 'Supreme', 'three', 'dozens', 'shortly', '2017', 'Defendants', 'campaign', 'attempt']

Cluster 3:
['helped', 'causes', 'argued', 'phone', 'Rosenberg', 'litigious', 'electric', 'closed', 'less', 'Please', 'Mueller', 'chairman', 'winning', 'found', 'factually', 'parties', 'first', 'Organization', 'bank', 'Moscow']

(e) 2-means clusters for T2 (for the entire text)

Clustering of T2:

Cluster 2:
['FBI', "Mary Trump's", 'Defendants', 'Republican', "Mary Trump's", 'Michael Cohen', 'John Durham', 'New York', 'dozens', 'Vance', 'Michael Cohen', '26', 'Trump', 'Res', 'Plaintiff in Chief', 'Last year', 'Democrats', 'dozens', 'Trump', '2020']

Cluster 1:
['the Supreme Court', 'NEW YORK', 'one', 'Red Scare', 'Kremlin', 'last week', 'Cyrus R. Vance Jr.', 'Ethel Rosenberg', 'the years', 'Associated Press', 'Justice Department', 'the early 1970s', 'Stephen Gillers', 'three years', 'The New York Times', 'the Trump Organization', 'the Trump Organization', 'Stormy Daniels - Cohen', 'more than a year and a half', 'Vladimir Putin']

(f) 1 page discussion of of results, observations, issues

The results of clustering for T1 and T2 above are limited to only 20 strings so as to not overcrowd the document. In reality cluster sizes vary and are much larger than 20. The clustering obtained from the process above groups together words that may not have similar meanings in the english language. For example in T1: the word 'Please and 'Moscow' are put together in cluster 3.

This creates an issue of consistency in our clusters. Although these two words don't have much correlation themselves, the clustering that was done for T1 takes into account the context of each word in the text. The reason they appear in the same cluster could be because the context they are used in may have some similarities.

Another thing to note with the clustering of T1 is that it takes into account all tokens that appear in the text document. A lot of these words add no value to our clustering but are still included. In a sense, these words such as punctuation words or conjunctions contribute to noise in our K means clustering. Cluster sizes grow very large because of these words which moves centroids around and this potentially gives rise to misclassifications.

When we consider only words of value (named entities) in our clustering we obtain much better results. Although not perfect the clustering makes much more sense in T2.

Using a relatively small number of clusters (3 and 2) for a text extract of large size also gives rise to misclassifications. Words become more generalized. Using a more clusters will give rise to clusters that are more specific in nature. Another technique that can be used is the elbow method for Kmeans clustering.

The elbow method is a way to figure out the most optimal value for K.
[More details follow in assignment 2]