Crystal Mosley
MSDS 453 Sec 55

Final Project – Trump Administration

Introduction

This project encompasses 38 documents (DSI's) all relating to the Trump Administration from reputable news sources with topics such as foreign affairs, immigration, the border wall, and selection of the new Supreme Court Justice – to name a few. The purpose of this analysis is to go through different natural language processing techniques such as: comparing manual and automated term extraction engines, coming up with reference term vectors (RTV's), classes for classification, clusters and similar topics between all the documents, and creating an ontology.

Term Extraction

The first objective in the process was to extract terms (nouns and noun phrases) to get an understanding of the concepts of the text within each document. Of my two documents, the manual extractions were very different (except for one extraction engine matching with Syria) than both the automated term extraction engines (TerMine and FiveFilters) as can be seen in Table 1. TerMine found all phrases versus single words, but still fall in line with FiveFilters and Manual extractions relating to Syria and Iran discussions. All three extraction processes found some form of the following in document 1: "trump", "administration", and "trump administration". "Trump administration" is the phrasing that I choose to utilize because "trump" by itself could also mean to fool someone/audience, and "administration" by itself could mean supervising a process. Document 1 also contains the following words and phrases that could relate to each other: "policy", "foreign policy", "foreign policy objective", and "policy objective". I don't believe these 4 could be lumped into one phrase due to how each could have differences in definition. Within Document 2, "U.S." and "United States" were found between both the manual extraction and Five Filters. Either words could be utilized as they both mean the same, United States.

Table 1: Term Extractions

Term/Phrase	Manual	Engine
syria	7	6

iran	7	
administration	7	
policy	6	
u.s.	7	
united nation general assembly		8
senior trump administration official		8
syrian president bashar al-assad		8
hagel		5

Reference Term Vectors (RTV's)

Next in the process was to decide on what the most prevalent terms were within my own documents as well as across the entire cohort of 38 documents, as well as compute the TF-IDF scores. TF-IDF evaluates how important a word or phrase is to a document in a collection, or corpus. The importance increases equally to the number of times a word/phrase appears in a document.

Experiment 1 – I used the word stemmer for this first round against the class corpus without any other changes.

Table 2: DSI 1 TF-IDF Scores (DNE = does not exist)

DSI 1 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Trump	0.15436	0.08669
US	0.03926	DNE
Syria	0	0.29415
Iran	0.0298	0

Table 3: DSI 2 TF-IDF Scores (DNE = does not exist)

DSI 2 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Trump	0.17915	0.05748
US	0.03674	DNE
Syria	0	0.21671
Iran	0.0298	0

The results of manual vs. python came out very different. Python found all lower scores than my manual scoring, and "US" did not come up at all in the data matrix. I did see other variations of "Iran", "US", and "Iran" that look more like two words put together as one, such as "iranalign", "iranback", and "iranianback". This could be because of the word stemming.

Experiment 2 – I removed the word stemmer for this next round against the class corpus without any other changes.

Table 4: DSI 1 TF-IDF Scores (DNE = does not exist)

DSI 1 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Trump	0.15436	0.08675
US	0.03926	DNE
Syria	0	0.28688
Iran	0.0298	0

Table 5: DSI 2 TF-IDF Scores (DNE = does not exist)

DSI 2 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Trump	0.17915	0.05834
US	0.03674	DNE
Syria	0	0.21436
Iran	0.0298	0

The results for this round were nearly the same for the python scores, with the only differences being that the "Trump" score was a little higher for this round, while the "Syria" score was a little lower than the previous round. I was surprised to find that the same two-word phrases were still in existence, even without the word stemming.

Experiment 3 – I tried other terms that were prevalent within both of my DSI's that also had Python scores without word stemming. I found that using word stemming caused these same words to not show as their full names. Important or prevalent terms within both DSI's: Policy, Administration and Tehran. Estimates for the number of documents within the class corpus in which each term appears: Policy – 14, Administration – 21 and Tehran – 6.

Table 6: DSI 1 TF-IDF Scores

DSI 1 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Policy	0.11512	0.18368
Administration	0.17255	0.07833
Tehran	0.0445	0.04255

Table 7: DSI 2 TF-IDF Scores

DSI 2 Word/Phrase	Manual TF-IDF Score	Python TF-IDF Score
Policy	0.08201	0.03431
Administration	0.11504	0.08779
Tehran	0.0445	0.19074

The results of this third round with different terms were interesting in that my manual score for DSI 1 "policy" was smaller than the python score, but the other two terms had a lower Python score than my manual. In

DSI 2, my manual score for “Tehran” was smaller than Python’s score, while the other two terms were lower for Python versus my manual scoring.

Classification

Next, I assessed the classification and clusters of the DSI’s. To process the text within the class corpus, I needed to create a function to handle the below actions:

1. Split document into individual words – tokenize
2. Remove punctuation from each word
3. Remove remaining tokens that are not alphabetic
4. Filter out short tokens
5. Lowercase all words
6. Filter out stop words
7. Word stemming - stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma.

Originally, I did a manual classification by reviewing each document title and skimming through the articles to try and summarize in one word or phrase, what I believed each should be classified as – shown in Table 8. I then grouped the documents together with other potential documents that were classified the same. When I ran the *tfidfvectorizer()* I found that *tweets, scandals, GDP, foreign policy, foreign affairs,* and *brett kavanaugh supreme court* were not found within the tf-idf matrix for scores and only 63% of the documents had a TF-IDF score.

Table 8: Original Classification of Class Corpus

Document	Classification	TF-IDF Score
PCS_Doc2_Trump-Widens-Availability-Short-Term-Health-Plans.docx	healthcare	0.04431
NM_Doc2_Trump_s_Grand_Plan_for_Healthcare.docx	healthcare	0.05507
RT_Doc1_Corp_Execs_describe_trump_mgmt_style_no_meta.docx	Trump	0.11768
RJB_Doc2_Canada_US_Reach_NAFTA_Deal.docx	trade	0.13636
EYM_Doc2_What-Trumps-Family.docx	immigration	0.15389
RJB_Doc1_Mexican_President_Insists_Trilateral_NAFTA.docx	trade	0.16782
PKC_Doc2_More-US-Canada-NAFTA.docx	trade	0.17088
RT_Doc2_trump_leadership_qualities_no_meta.docx	Trump	0.19059
PKC_Doc1_US-Canada-NAFTA.docx	trade	0.20954
JMK_Doc2_Donald-Trump-agrees-to-cease-fire-in-the-trade-war-with-the-EU.docx	trade	0.23412
JMK_Doc1_Is-China-losing-the-trade-war-against-America.docx	trade	0.25937
AC_Doc2_Trump-Trade-China.docx	trade	0.32854
AC_Doc1_Trump-Trade-S-Korea.docx	trade	0.36795
EYM_Doc1_Trumps-Immigration-Reversal.docx	immigration	0.42215

I decided to go back through the python code and changed the algorithm to run without the *porterstemmer()*. I noticed the matrix dataframe went from having 2,232 words, to having 2,945 words. I then updated the classifications more in line with the system generated clusters (using k-means = 8) and found new tf-idf scores for the tops terms, shown in **Table 9**.

Table 9: Updated Classifications of Class Corpus

Document	Classification
AK_Doc1_President_Donald_Trump_praised.docx	Judge Brett Kavanaugh
AK_Doc2_ Senate_Republicans_have_pushed.docx	Judge Brett Kavanaugh
ATS_Doc1_White-House-Tells.docx	Judge Brett Kavanaugh
ATS_Doc2_Yale-Classmate-to-Tell.docx	Judge Brett Kavanaugh
NPL_Doc1_Trump-Defends-Kavanaugh.docx	Judge Brett Kavanaugh
NPL_Doc2_Trump-Mocks-Kavanaugh-Accuser.docx	Judge Brett Kavanaugh
TAC_Doc1_Republicans_Name_Democrats_Mob.docx	Judge Brett Kavanaugh
TAC_Doc2_Kavanaugh_Confirmation_Vote.docx	Judge Brett Kavanaugh
OE_Doc1_Team-Trump-Paid.docx	Judge Brett Kavanaugh
AC_Doc1_Trump-Trade-S-Korea.docx	trade
AC_Doc2_Trump-Trade-China.docx	trade
JMK_Doc1_Is-China-losing-the-trade-war-against-America.docx	trade
JMK_Doc2_Donald-Trump-agrees-to-cease-fire-in-the-trade-war-with-the-EU.docx	trade
JMD_Doc2_4.1-GDP-Growth.docx	trade
MET_Doc1_Transcript-Press-Conference.docx	trade
RT_Doc1_Corp_Execs_describe_trump_mgmt_style_no_meta.docx	Trump
RT_Doc2_trump_leadership_qualities_no_meta.docx	Trump
OE_Doc2_Melania-Staying-in-Hotel.docx	Trump
JMD_Doc1_Tweet-Had-Extra-Zero.docx	Trump
MET_Doc2_Trump_Tweets.docx	Trump
CMM_Doc1_Its-Complete-Folly_googleDrive.docx	foreign policy
CMM_Doc2_Trump-Officials-Target_googleDrive.docx	foreign policy
HJC_DSI1_US_expects_to_eliminate_Iranian_oil_imports.docx	foreign policy
HJC_DSI2_Trump's_battering_of_Iran_the_cost_of_oil.docx	foreign policy
SF Rouhani Trump administration most 'spiteful' to Iran in 40 years.docx	foreign policy
SF Trump, Iran_s Rouhani exchange threats, insults on U.N._s world stage.docx	foreign policy
PKC_Doc1_US-Canada-NAFTA.docx	NAFTA
PKC_Doc2_More-US-Canada-NAFTA.docx	NAFTA
RJB_Doc1_Mexican_President_Insists_Trilateral_NAFTA.docx	NAFTA
RJB_Doc2_Canada_US_Reach_NAFTA_Deal.docx	NAFTA

JEN_Doc1_Russia-has-never-interfered.docx	election interference
JEN_Doc2_CohortX-US-Society-based-on-leaks.docx	election interference
EYM_Doc1_Trumps-Immigration-Reversal.docx	immigration
EYM_Doc2_What-Trumps-Family.docx	immigration
NM_Doc1_Trumps_Plan_to_reshap_healthcare.docx	healthcare
NM_Doc2_Trump_s_Grand_Plan_for_Healthcare.docx	healthcare
PCS_Doc1_Trump-Health-Insurance-Policies-Headwinds.docx	healthcare
PCS_Doc2_Trump-Widens-Availability-Short-Term-Health-Plans.docx	healthcare

The new clustering output matched with about 82% of my classifications. The exceptions were:

- OE_Doc2_Melania-Staying-in-Hotel.docx – Trump
    - I agree with this change from my second review of classifying this document as judge brett kavanaugh. This is because the article has nothing to do with Brett Kavanaugh and has more to do with Trump and his personal life. This is also not a scandal as I had originally classified.
- JMD_Doc1_Tweet-Had-Extra-Zero.docx – Trump
    - I agree with this change from tweets to Trump since this has more to do with Trump and his social media mistake.
- MET_Doc2_Trump_Tweets.docx – Trump
    - The system classified this document as part of the NAFTA group classifications; I do not agree with this change. I changed it from tweets to Trump since this has more to do with Trump and his social media mistake.

The below classifications changed from trade to NAFTA. I agree with this change since each article is specific to NAFTA rather than just trade and trade agreements:

- PKC_Doc1_US-Canada-NAFTA.docx - NAFTA
- PKC_Doc2_More-US-Canada-NAFTA.docx - NAFTA
- RJB_Doc1_Mexican_President_Insists_Trilateral_NAFTA.docx - NAFTA
- RJB_Doc2_Canada_US_Reach_NAFTA_Deal.docx – NAFTA

After final review, I decided to change JEN_doc1 and JEN_doc2 from foreign affairs to election interference as both articles discuss the Russian election interference, specifically.

Clusters

Next, I evaluated multiple clusters using the K-Means Clustering algorithm with three different k-means: 5, 8, and 10; I also utilized the *word2vec()* and *doc2vec()* to experiment with the class DSI's. A Word2Vec effectively captures semantic relations between words, hence, can be used to calculate word similarities or fed as features to various NLP tasks such as sentiment analysis. Words can only capture so much, there are times when you need relationships between sentences and documents and not just words, which is where the Doc2Vec comes into play. Out of the three, k-means 5 and 10 with the Word2Vec algorithm worked best as shown in **Figures 1 and 2**. With

k=5, four clusters were predominant and close to the five clustering that I was looking for. With k=10, five clusters were predominant and two other smaller cluster were a little close together, but overall, it stayed more towards to five clusters that I started with.

Figure 1: K=5 Cluster with Word2Vec Results

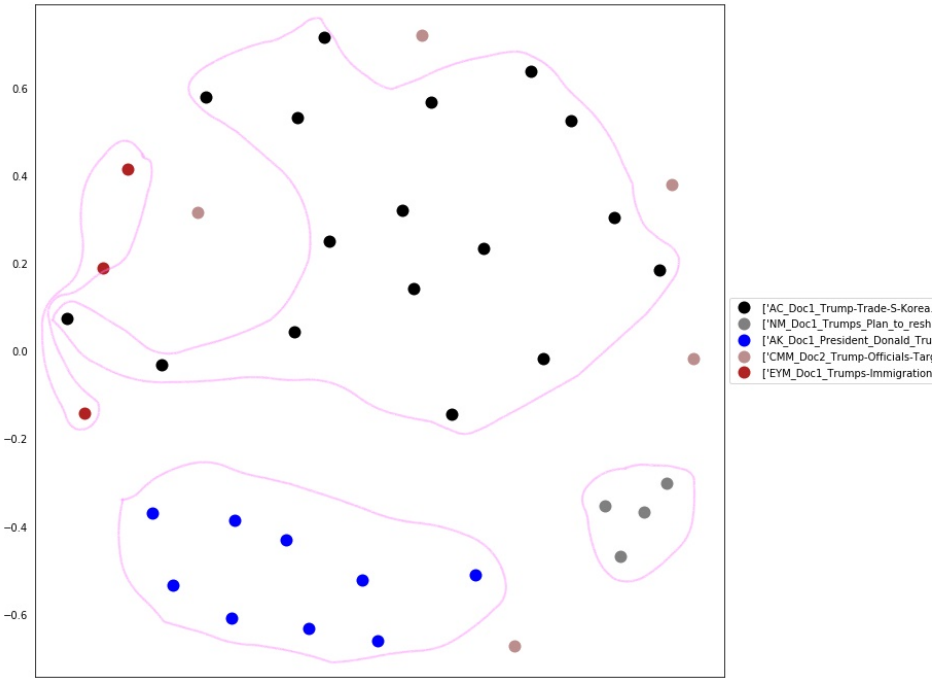
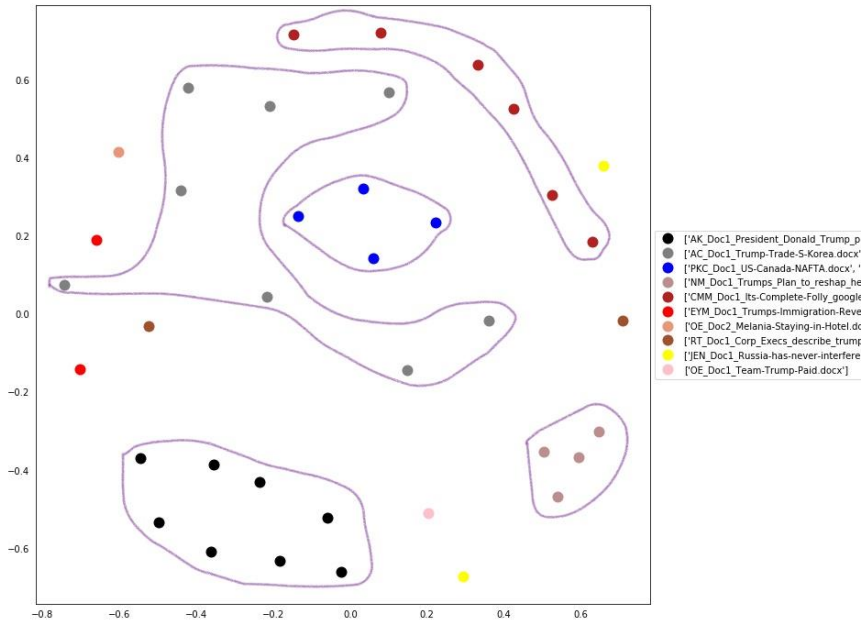


Figure 1: K=10 Cluster with Word2Vec Results



The documents clustered great with k=10, as shown in **Table 10**, but show obvious signs of misclassification from my manual classes versus system classes. The classes were changed in the previous section to match the system classifications.

Table 10: Manual Classifications with k=10 Clustering Results

AK_Doc1_President_Donald_Trump_praised.docx	brett kavanaugh supreme court	cluster 3
AK_Doc2_ Senate_Republicans_have_pushed.docx	brett kavanaugh supreme court	cluster 3
ATS_Doc1_White-House-Tells.docx	brett kavanaugh supreme court	cluster 3
ATS_Doc2_Yale-Classmate-to-Tell.docx	brett kavanaugh supreme court	cluster 3
NPL_Doc1_Trump-Defends-Kavanaugh.docx	brett kavanaugh supreme court	cluster 3
NPL_Doc2_Trump-Mocks-Kavanaugh-Accuser.docx	brett kavanaugh supreme court	cluster 3
TAC_Doc1_Republicans_Name_Democrats_Mob.docx	brett kavanaugh supreme court	cluster 3
TAC_Doc2_Kavanaugh_Confirmation_Vote.docx	*brett kavanaugh supreme court*	*N/A*
JEN_Doc1_Russia-has-never-interfered.docx	foreign affairs	cluster 7
JEN_Doc2_CohortX-US-Society-based-on-leaks.docx	foreign affairs	cluster 7
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
HJC_DSI1_US_expects_to_eliminate_Iranian_oil_imports.docx	foreign policy	cluster 5
HJC_DSI2_Trump's_battering_of_Iran_the_cost_of_oil.docx	foreign policy	cluster 5
MET_Doc1_Transcript-Press-Conference.docx	foreign policy	cluster 9
SF Rouhani Trump administration most 'spiteful' to Iran in 40 years.docx	foreign policy	cluster 5
SF Trump, Iran_s Rouhani exchange threats, insults on U.N._s world stage.docx	foreign policy	cluster 5
JMD_Doc1_Tweet-Had-Extra-Zero.docx	GDP	cluster 4
JMD_Doc2_4.1-GDP-Growth.docx	GDP	cluster 4
NM_Doc1_Trumps_Plan_to_reshap_healthcare.docx	healthcare	cluster 2
NM_Doc2_Trump_s_Grand_Plan_for_Healthcare.docx	healthcare	cluster 2
PCS_Doc1_Trump-Health-Insurance-Policies-Headwinds.docx	healthcare	cluster 2
PCS_Doc2_Trump-Widens-Availability-Short-Term-Health-Plans.docx	healthcare	cluster 2
EYM_Doc1_Trumps-Immigration-Reversal.docx	immigration	cluster 8
EYM_Doc2_What-Trumps-Family.docx	immigration	cluster 8
OE_Doc1_Team-Trump-Paid.docx	scandals	cluster 4
OE_Doc2_Melania-Staying-in-Hotel.docx	scandals	cluster 4
AC_Doc1_Trump-Trade-S-Korea.docx	trade	cluster 9
AC_Doc2_Trump-Trade-China.docx	trade	cluster 9
JMK_Doc1_Is-China-losing-the-trade-war-against-America.docx	trade	cluster 9
JMK_Doc2_Donald-Trump-agrees-to-cease-fire-in-the-trade-war-with-the-EU.docx	trade	cluster 9
PKC_Doc1_US-Canada-NAFTA.docx	trade	cluster 0
PKC_Doc2_More-US-Canada-NAFTA.docx	trade	cluster 0
RJB_Doc1_Mexican_President_Insists_Trilateral_NAFTA.docx	trade	cluster 0
RJB_Doc2_Canada_US_Reach_NAFTA_Deal.docx	trade	cluster 0
RT_Doc1_Corp_Execs_describe_trump_mgmt_style_no_meta.docx	Trump	cluster 6
RT_Doc2_trump_leadership_qualities_no_meta.docx	Trump	cluster 6
MET_Doc2_Trump_Tweets.docx	tweets	cluster 9

Below are the top 10 key terms that characterize each of the clusters:

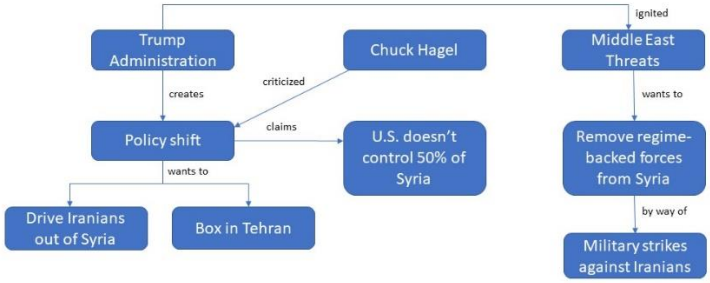
Cluster 0:	Cluster 1:	Cluster 2:	Cluster 3:	Cluster 4:	Cluster 5:	Cluster 6:	Cluster 7:	Cluster 8:	Cluster 9:
canada	syria	insur	senat	trump	rouhani	leadership	russian	famili	trade
agreement	iranian	plan	kavanaugh	melania	sanction	confid	russia	immigr	china
nafta	hagel	shortterm	investig	capit	price	dimens	putin	separ	great
trade	bolton	health	republican	growth	trump	manag	elect	order	south
congress	troop	peopl	trump	clifford	nuclear	trump	polit	reunit	tariff
mexico	middl	coverag	judg	alleg	regim	candid	interfer	children	percent
unit	tehran	obamacar	nomin	dailymailcom	iranian	survey	lavrov	parent	chines
canadian	foreign	afford	alleg	presid	nation	particip	foreign	detain	korea
state	forc	cover	confirm	unemploy	unit	rate	presidenti	trump	korean
negoti	fighter	consum	suprem	report	offici	expert	relat	execut	steel

Ontology

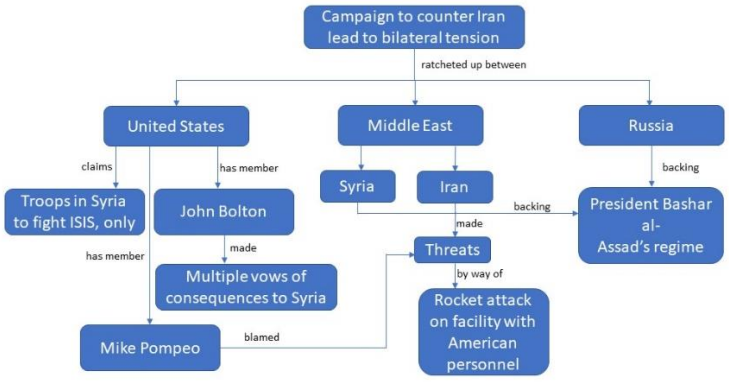
Finally, I went through an ontology process. An ontology encompasses a representation, formal naming, and definition of the categories, properties, and relations between the concepts, data, and entities that substantiate one, many, or all domains.

At first pass, I thought my DSI 1 ("Its-Complete-Folly") was easy to build and DSI 2 ("Trump-Officials-Target") was tough to build – reason being, DSI 2 had a lot going on in many directions and I thought it was more like a spider web; however, DSI 2's ontology came out as expected, and DSI 1 was too simplified.

CMM_Doc1_Its-Complete-Folly



CMM_Doc2_Trump-Officials-Target



Conclusion

Of the entire process of analyzing the class corpus, the most difficult task was creating the classes for classifying each document. For example, I classified my documents as *foreign policy*, but it could have easily been classified as *foreign affairs* or more specifically, *sanctions*, since both documents specifically talked about the Trump administration putting sanctions on the Middle East. The same occurred for the DSI's that I classified as *trade*; those documents were specific to *NAFTA*, so their classifications changed to NAFTA in the end because it made more sense. Ontologies were a headache for me in the beginning because they reminded me of an extreme version of ER diagrams, only more difficult because I had to go through the article and pinpoint what each entity topic should start as, and any attributes that they may be related to. Now that I have gone through the ontology process, I believe I have a better understanding of how to create them and find that articles with more going on are better to work with.