Assignment 1.

Vectorized Representation

MSDS 453: Natural Language Processing

Husein Adenwala

Northwestern University

1. Introduction & Problem Statement

The purpose of this assignment is to compare and evaluate different vectorization and word embedding techniques on the 249 manually collected and labeled movie reviews. The objective is to understand which technique and parameters are best suited for creating a vocabulary for the given corpus which will eventually be used for sentiment analysis. This research will experiment with different data wrangling methods to create TF-IDF vectors and Doc2Vec and Word2Vec embedding sizes of varying lengths. Finally, I will use a cosine similarity matrix, T-SNE plot and k-means clustering to evaluate the vectorizing and embedding created by all the experiments.

2. Literature review

TF-IDF, Word2Vec, and Doc2Vec are frequently used Word Embedding methods. TF-IDF reflects the relative importance of a word over the whole corpus in a text. Term frequency (TF) is the total number of times a term(t) exists in a document(D) and the inverse document frequency (IDF) is the reciprocal number of documents(d) in which the term appears over the total number of documents(N). The importance of the word increases as the increase in the number of times it appears in a document, but it is neutralized by the frequency of the word in the entire data set. The function calculates the tf – idf as tf-idf = tf(t, D) * idf(d, N). A high tf-idf score implies a high frequency of a word in a document and a low frequency of documents for that term over the entire document collection (Jitendra, 2021).

Word2vec and doc2vec techniques are the new techniques for preserving the whole corpus of contextual word knowledge. It basically provides a dense vector representation of a term which has a semantic meaning (Mikolov et al, 2013).

Vectorized Representation

Doc2vec, a numerical representation of a document, is a modified form of embedding word2vec for a large set of text including paragraphs or documents. While the word vector depicts a word, the document vector offers the definition of a document (Mikalov et al, 2013). The document-ID is also trained while training the word vectors, and it retains a numeric representation of the document at the end of the training. Doc2vec model representation makes the algorithm quicker, and less memory consuming. The current analysis resulted in a better performance by using the document vector representation in the machine learning model. There is not a unique technique that dominates the others, because each one has a better behavior for each type of content, or according to each use case (Aguilar et al, 2020).

3. Data preparation, exploration, visualization

The data consists of 249 movie reviews of 25 movies that was stored and uploaded in a csv file. The tables and plots below show the movies' names, the number of movie reviews and movie genres.

All 25 movies have 10 reviews except the movie Martian and Red Notice, which have 9 and 20 reviews respectively. The number of positive and negative reviews are equal for all movies except Martian. Also, it is important to note that the count for each movie genre is unbalanced; the Drama genre has only 9 reviews.

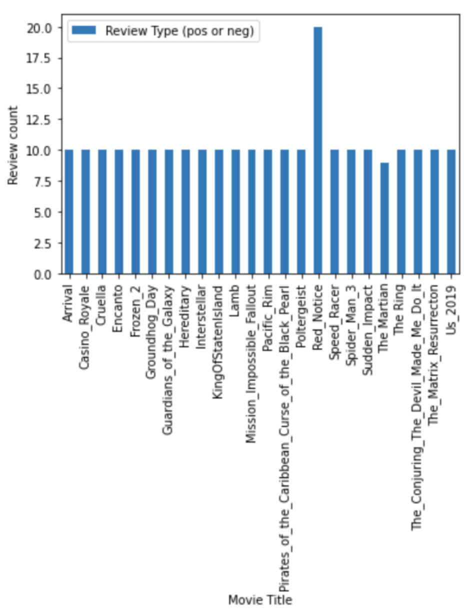
Vectorized Representation

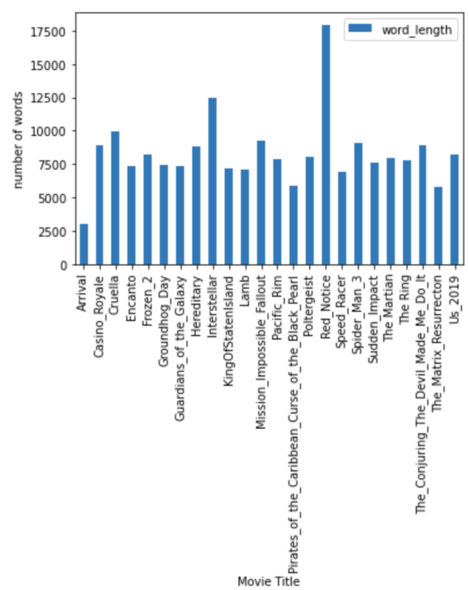
Movie Name	Number of Reviews
Arrival	10
CasinoRoyale	10
Cruella	10
Encanto	10
Frozen2	10
GroundhogDay	10
GuardiansOfTheGalaxy	10
Hereditary	10
Interstellar	10
KingOfStatenIsland	10
Lamb	10
MissionImpossibleFallout	10
PACIFICRIM	10
PiratesOfTheCaribbean:TheCurseOfTheBlackPearl	10
PiratesOfTheCarribean	10
Poltergeist	10
RedNotice	20
SpeedRacer	10
SpiderMan3	10
TheConjuring3	10
TheMartian	9
TheMatrixResurrecton	10
TheRing	10
Us	10

Genre of Movie	Number of Reviews
Action	70
Comedy	60
Drama	9
Horror	60
Sci_Fi	50

```
Total word count:  198977
```

Movie Title	Review Type (pos or neg)	
Arrival	Negative	5
	Positive	5
Casino_Royale	Negative	5
	Positive	5
Cruella	Negative	5
	Positive	5
Encanto	Negative	5
	Positive	5
Frozen_2	Negative	5
	Positive	5
Groundhog_Day	Negative	5
	Positive	5
Guardians_of_the_Galaxy	Negative	5
	Positive	5
Hereditary	Negative	5
	Positive	5
Interstellar	Negative	5
	Positive	5
KingOfStatenIsland	Negative	5
	Positive	5
Lamb	Negative	5
	Positive	5
Mission_Impossible_Fallout	Negative	5
	Positive	5
Pacific_Rim	Negative	5
	Positive	5
Pirates_of_the_Caribbean_Curse_of_the_Black_Pearl	Negative	5
	Positive	5
Poltergeist	Negative	5
	Positive	5
Red_Notice	Negative	10
	Positive	10
Speed_Racer	Negative	5
	Positive	5
Spider_Man_3	Negative	5
	Positive	5
Sudden_Impact	Negative	5
	Positive	5
The Martian	Negative	5
	Positive	4
The Ring	Negative	5
	Positive	5
The_Conjuring_The_Devil_Made_Me_Do_It	Negative	5
	Positive	5
The_Matrix_Resurrecton	Negative	5
	Positive	5
Us_2019	Negative	5
	Positive	5





Vectorized Representation

Most common word in the text corpus:

Data Processing

All movie reviews were combined to create the Corpus. This corpus was tokenized and normalized. I experimented with 5 methods for normalizing the data such as removing punctuation, stop word removal, stemming, lemmatization, lower case, key term extraction (NTLK Rake()) and removing non-alphabet words.

The Corpus was then tokenized to create the vocabulary that was used to create TF-IDF and Doc2Vec and Word2Vec embeddings. I experimented with Doc2Vec and Word2Vec embedding sizes of 100, 200 and 300.

4. Research Design and Modeling Method(s)

Step 1 Qualitative approach

I manually extracted key terms that I found important and prevalent in my movie review documents (Frozen 2). I found cast names, movie names and character names to be important and adjectives, verbs and catch phrases that are commonly used to describe movies to be prevalent.

Vectorized Representation

Following is the key term extraction from my 10 reviews of Frozen 2:

```
['Frozen 2', 'Elsa', 'Anna', 'Kristoff', 'Sven', 'Olaf', 'Josh Gad', 'Kristen Bell', 'Jonathan Groff', 'Idina Menzel', 'spectac
ular', 'sweet spot', 'fun', 'cash grab', 'warm', 'disappointed', 'bored', 'joy', 'lackluster', 'sweet', 'jaded', 'enjoy', 'Menz
el', 'disappointed', 'abhorent', 'sequelentertain', 'Frozen', 'comedy', 'surprising', 'underwhelmed', 'dazzling', 'pleasant']
```

Step 2 Quantitative approach

I evaluated 5 data wrangling method to understand the effect on the vocabulary and its subsequent effect on vectorizing and embeddings. For each of the data wrangling methods, I also evaluated the impact of changing vector size for Doc2Vec and Word2Vec.

The following table describes the 23 experiments.

	Method 1			**Method 2**			**Method 3**			**Method 4**	**Method 5**
Data Wrangling Methods	Remove punctuations			Remove punctuation , stop word , stemming and lower case			Remove punctuation and stop word and lemmatization, remove non alphabet and			remove punctuation and ngrams = 2	remove punctuation and extract key term using Rake() function (NTLK)
Word2Vec embedding vector size	100	200	300	100	200	300	100	200	300	300	300
Doc2Vec embedding vector size	100	200	300	100	200	300	100	200	300	300	300

Methods 1-5 created a vocabulary size of approximate 17,400; 11,400; 15,400; 128,500 and 17,300 words respectively.

The goal in methods 1-3 was to progressively remove non-important words; however, in method 2, I found stemming was incorrectly modifying words. For example, movie was stemmed to movi. I also, evaluated the performance of ngams (2) technique and Key term extraction technique using NTLK Rake library that is a domain-independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurrence with other words in the text.

I also experimented with the impact of Doc2Vec and Word2Vec embedding vector size in capturing the semantic meaning in the word and document of the text corpus.

To evaluate the different data wrangling and embedding vector sizes, I created T-SNE plot and did K-mean cluster analysis to analyze the vector space, where in theory, words that have similar meaning would be in the same vector space.

5. Results

Experiments:

23 experiments were done by varying data wrangling methods and Word2Vec and Doc2Vec wedding vector sizes.

Below are TF-IDF mean scores results, cosine similarity matrices and T-SNE plots for the 23 experiments.

	Method 1		
Data Wrangling Methods	Remove punctuations		
Word2Vec embedding vector size	100	200	300
Doc2Vec embedding vector size	100	200	300

Words match with Step 1 manual extraction

	word	mean_tfidf_score
0	Frozen	1.209
1	fun	1.035
2	Elsa	1.010
3	comedy	0.889
4	Anna	0.613
5	enjoy	0.358
6	sweet	0.293
7	spectacular	0.289
8	Kristoff	0.282
9	Olaf	0.248
10	Menzel	0.237
11	disappointed	0.237
12	warm	0.182
13	joy	0.182
14	surprising	0.156
15	Sven	0.139
16	dazzling	0.139
17	bored	0.110
18	lackluster	0.095
19	pleasant	0.095
20	frozen	0.079
21	jaded	0.062
22	underwhelmed	0.023

Top 10 tf-idf mean scores

	word	mean_tfidf_score
0	the	40.414
1	and	22.426
2	of	21.631
3	to	19.458
4	is	12.665
5	in	12.582
6	that	10.895
7	it	8.013
8	with	7.010
9	as	6.771

	Method 2		
Data Wrangling Methods	Remove punctuation , stop word , stemming and lower case		
Word2Vec embedding vector size	100	200	300
Doc2Vec embedding vector size	100	200	300

Words match with Step 1 manual extraction

	word	mean_tfidf_score
0	elsa	1.308
1	frozen	1.195
2	fun	1.037
3	anna	0.729
4	enjoy	0.636
5	sweet	0.345
6	kristoff	0.315
7	spectacular	0.299
8	olaf	0.265
9	menzel	0.254
10	joy	0.245
11	warm	0.195
12	sven	0.139
13	pleasant	0.095

Top 10 tf-idf mean scores

	word	mean_tfidf_score
0	film	5.292
1	movi	4.938
2	like	3.271
3	bond	2.974
4	cruella	2.853
5	make	2.518
6	famili	2.435
7	time	2.405
8	charact	2.300
9	stori	2.118

	Method 3		
Data Wrangling Methods	Remove punctuation and stop word and lemmatization, remove non alphabet and		
Word2Vec embedding vector size	100	200	300
Doc2Vec embedding vector size	100	200	300

Words match with Step 1 manual extraction

	index	word	mean_tfidf_score
0	5504	frozen	1.179
1	5527	fun	1.037
2	4360	elsa	1.010
3	2545	comedy	0.952
4	540	anna	0.729
5	4505	enjoy	0.358
6	12597	spectacular	0.299
7	13318	sweet	0.293
8	7576	kristoff	0.282
9	9272	olaf	0.248
10	8428	menzel	0.237
11	3752	disappointed	0.237
12	7385	joy	0.207
13	14800	warm	0.182
14	13246	surprising	0.156
15	3315	dazzling	0.152
16	13287	sven	0.139
17	1518	bored	0.110
18	9997	pleasant	0.095
19	7601	lackluster	0.095
20	7265	jaded	0.062
21	14240	underwhelmed	0.023

Top 10 tf-idf mean scores

	word	mean_tfidf_score
0	film	5.277
1	movie	4.938
2	like	3.133
3	bond	3.007
4	cruella	2.853
5	time	2.373
6	family	2.351
7	character	2.300
8	make	2.218
9	story	2.106

	Method 4
Data Wrangling Methods	remove punctuation and ngrams = 2
Word2Vec embedding vector size	300
Doc2Vec embedding vector size	300

Words match with Step 1 manual extraction

word	mean_tfidf_score
jonathan groff	0.182
kristen bell	0.169
idina menzel	0.156
josh gad	0.156
cash grab	0.129
sweet spot	0.062

Top 10 tf-idf mean scores

	word	mean_tfidf_score
0	the the	46.610
1	and and	23.430
2	of of	21.799
3	to to	19.614
4	in in	13.501
5	is is	12.786
6	that that	11.427
7	it it	10.275
8	as as	7.348
9	with with	7.219

	Method 5
Data Wrangling Methods	remove punctuation and extract key term using Rake() function (NTLK)
Word2Vec embedding vector size	300
Doc2Vec embedding vector size	300

Words match with Step 1 manual extraction

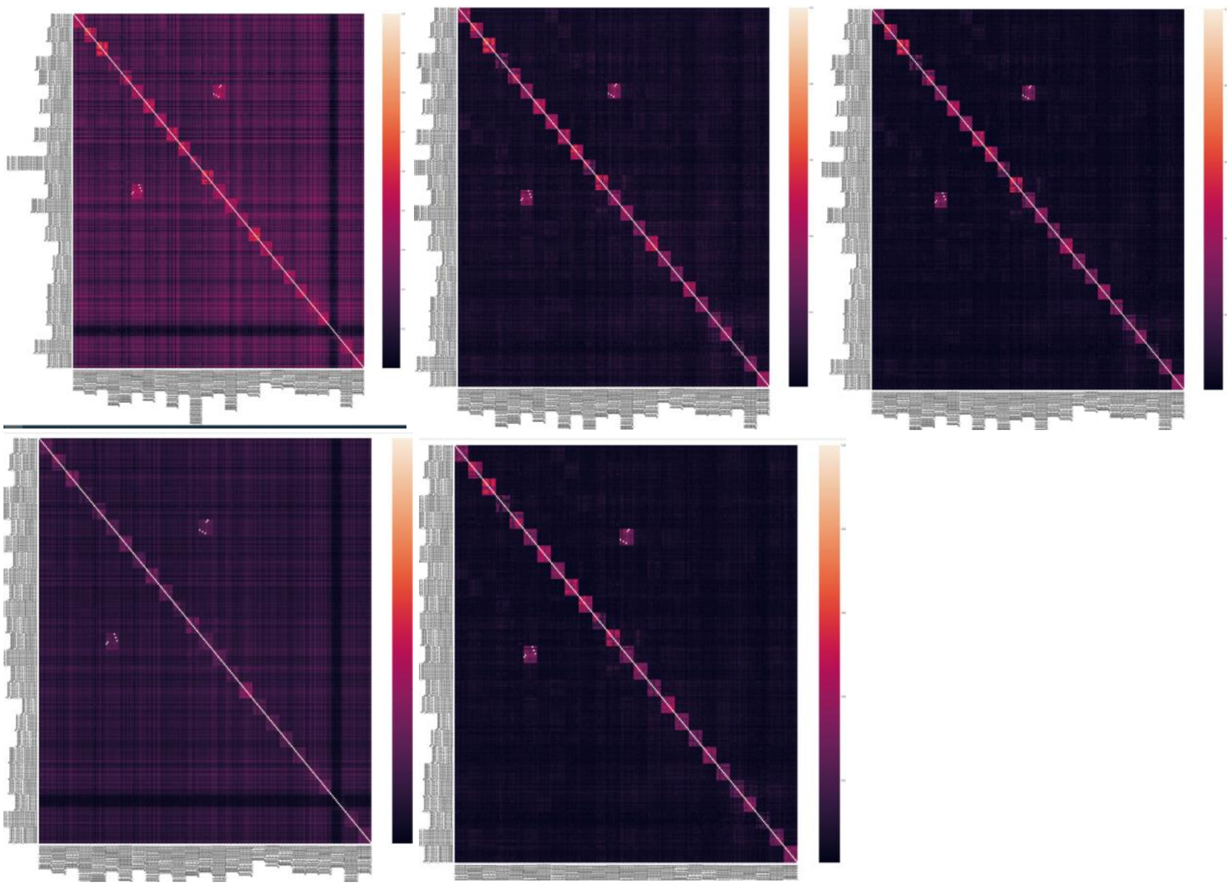
	word	mean_tfidf_score
0	frozen	1.179
1	fun	1.037
2	elsa	1.010
3	comedy	0.899
4	anna	0.613
5	enjoy	0.358
6	spectacular	0.299
7	sweet	0.293
8	kristoff	0.282
9	olaf	0.248
10	menzel	0.237
11	disappointed	0.237
12	joy	0.195
13	warm	0.182
14	surprising	0.156
15	dazzling	0.152
16	sven	0.139
17	bored	0.110
18	pleasant	0.095
19	lackluster	0.095
20	jaded	0.062
21	underwhelmed	0.023

Top 10 tf-idf mean scores

	word	mean_tfidf_score
0	film	4.548
1	movie	3.886
2	one	3.482
3	like	3.100
4	cruella	2.853
5	bond	2.782
6	family	2.234
7	time	2.183
8	harry	1.998
9	much	1.967

Vectorized Representation

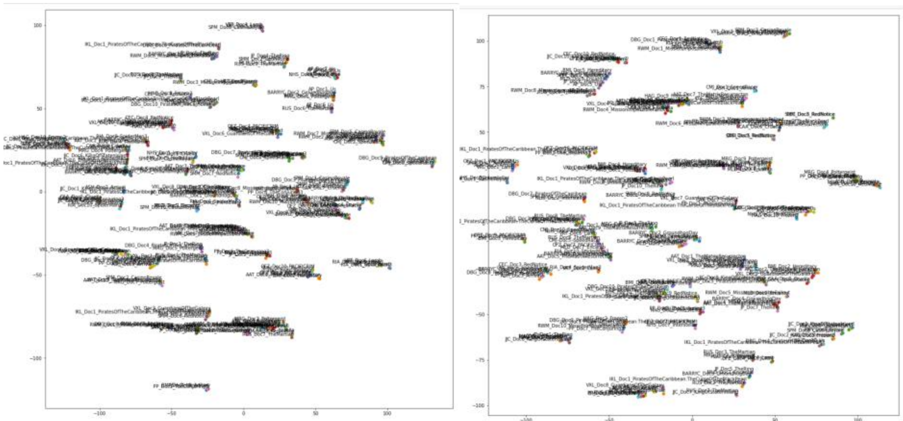
TF-IDF cosine similarity map for method 1, method 2, method 3, method 4 and method 5 respectively.



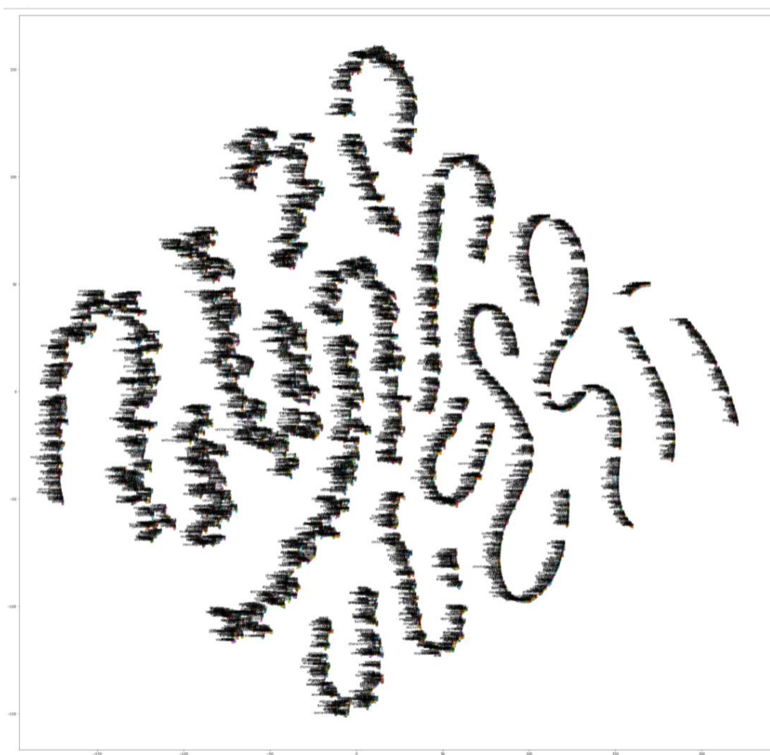
Data wrangling methods 3 and 5 produced the best result and were able to discriminate between vectors from different movies except there was overlap for the movie Red Notice (which was the movie that had 20 reviews from two different IDs).

Doc2Vec TSNE plot for methods 3 and 5, where the vector size was 300, provided distinct clusters as seen below.

Vectorized Representation



Word2Vec T-SNE plot for method 3 where Word2Vec embedding was 300 was marginally better compared to other wrangling methods and vector size.



Analysis and Interpretation

Vectorized Representation

Key Findings:

Comparing with words in Step1 and mean scores

As seen in the results above, Methods 1, 3, and 5 produced the best match with the manually selected words, the words that I thought would be important and prevalent. However, they did not have a high mean score compared to the more common words such as movie, character, like etc. The words that have high mean score are words that are the commonly used in movies reviews (word such as like, movie, film) and should be removed in the data wrangling process. This shows that data wrangling can further improved.

Data wrangling methods:

I found method 3 and method 5 of data wrangling most effective in reducing the noise from the corpus. Method 3 (i.e. Removing punctuation, stop word removal, lemmatization, lower case and removing non-alphabet words) and model 5 (using the NTLK Run Package (result similar to method 3)) reduced the noise in the data and produced the best Cosine similarly matrix, T-SNE and K-mean cluster plot.

Embedding vector size

Increasing vector size marginally improved the clustering in the T-SNE plot. A vector of size 300 produced the best Doc2Vec T-SNE that could distinguish between several clusters. T-SNE plot for Word2Vec did not show clusters in elongated lines which might suggest a connection between several words.

Best Method

Method 3 of data wrangling with embedding size of 300 for Doc2Vec and Word2Vec was the best at creating distinct clusters in the vector space as seen in the T-SNE plot and cosine similarity matrix heatmap. This was also reflected in K-means cluster analysis.

The elbow method for TF-IDF as seen in the plot below shows that there are probably 25 optimal clusters; however, that cannot be discriminated in the K-means cluster plot but intuitively makes sense as there are 25 movies.

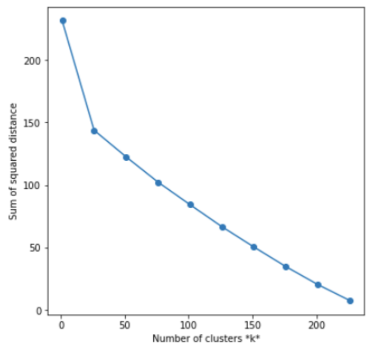
The elbow method for Doc2Vec as seen in the plot below shows that the idea cluster is between 3 and 6; however, there are 4 distinct clusters in K-means cluster plot. This could represent the 5 movie genres; however, as we have noted above, the representation of movie genres was unbalanced, as the Drama genre only had 9 reviews.

Vectorized Representation

K-mean analysis Method 3 – embedding vector size 300

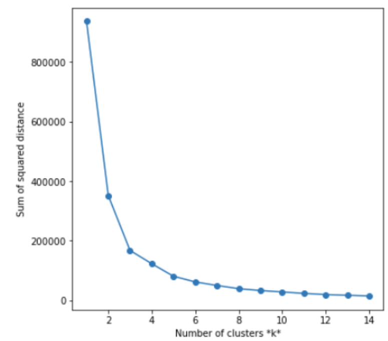
TF-IDF Matrix

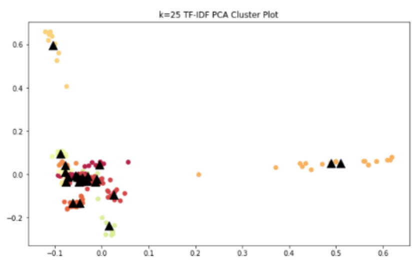
Elbow plot

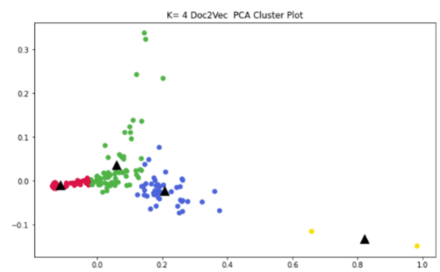


Doc2Vec matrix

Elbow plot







6. Conclusion

The results of the 23 experiments show that the data wrangling improves the vocabulary and creates vectors and embedding vectors that provide semantic meaning; however, the improvements were modest. I found method 3 to be the best data wrangling method in the experiment which normalized the data by removing punctuation, stop words, non-alphabet words; lemmatization and converting all the text to lower case. Also, I found that increasing 100, 200 to 300 slightly improves (at a diminishing rate) clustering in the T-SNE plot, which seems to suggest that the vectors are able to put similar-meaning words or documents in clusters; however, this might not be the case and requires further analysis.

In general, data wrangling has more of an impact on the quality of the vocabulary. The data wrangling methods that we have applied in this assignment are domain-independent and as a result retain many non-important words in the corpus such as movie, like, film etc. In the next steps, we need to apply data wrangling methods that are more apt for movie reviews.

References:

Aguilar, J., Salazar, C., Velasco, H., Monsalve-Pulido, J.A., & Montoya, E. (2020). Comparison and Evaluation of Different Methods for the Feature Extraction from Educational Contents. *Comput., 8*, 30.

Mikolov, T., Chen, K., Corrado, G.S., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ICLR*.

Jitendra, S. (2021). Natural Language Processing using Tfidf , Word2vec and Bert. Towards Data sciendce. https://medium.com/@js2441995/natural-language-processing-using-tfidf-word2vec-and-bert-825cc2c663c3

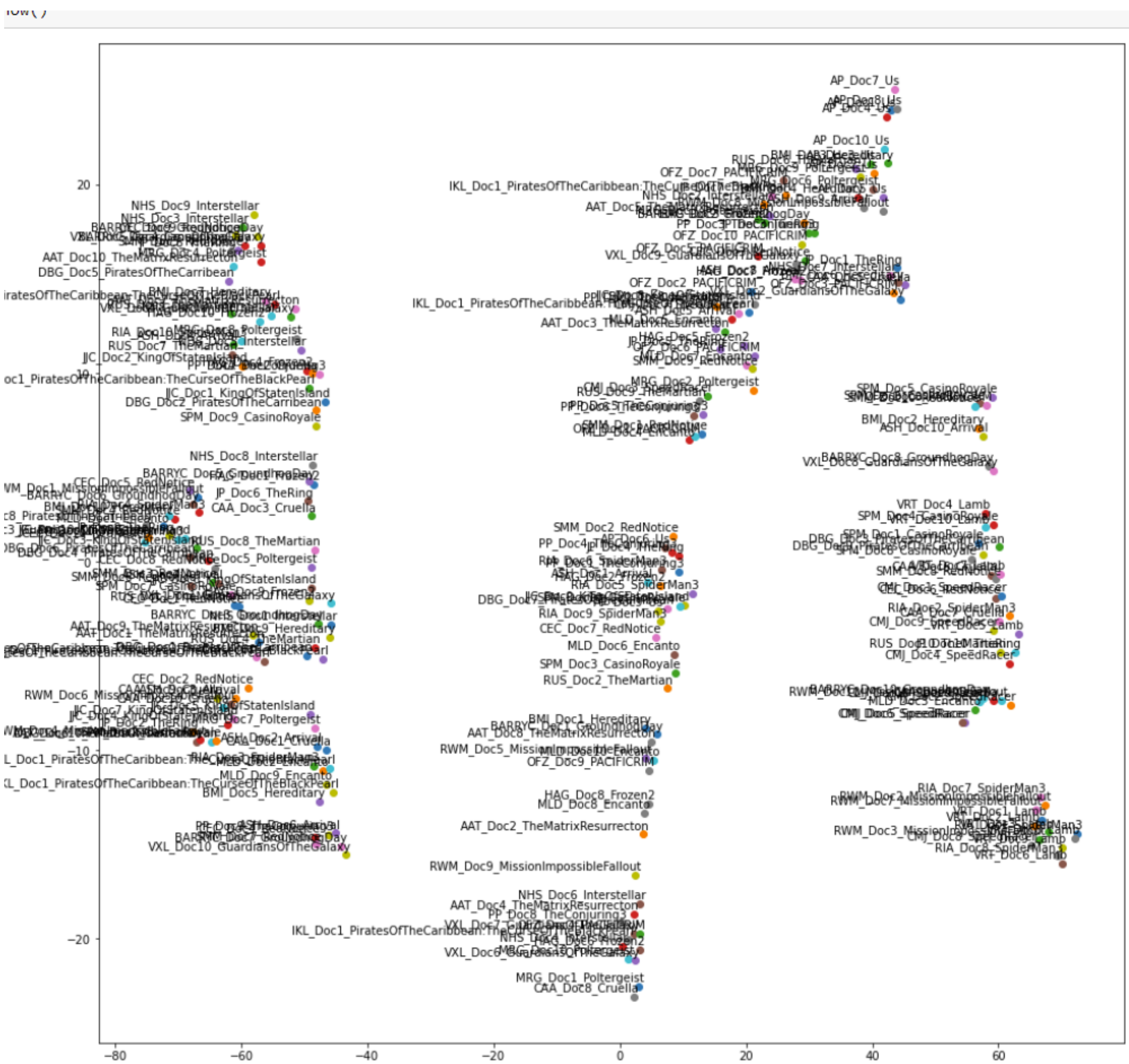
Appendix:

Appendix A

Method 3: Word2Vec Doc2Vec Embedding size 100

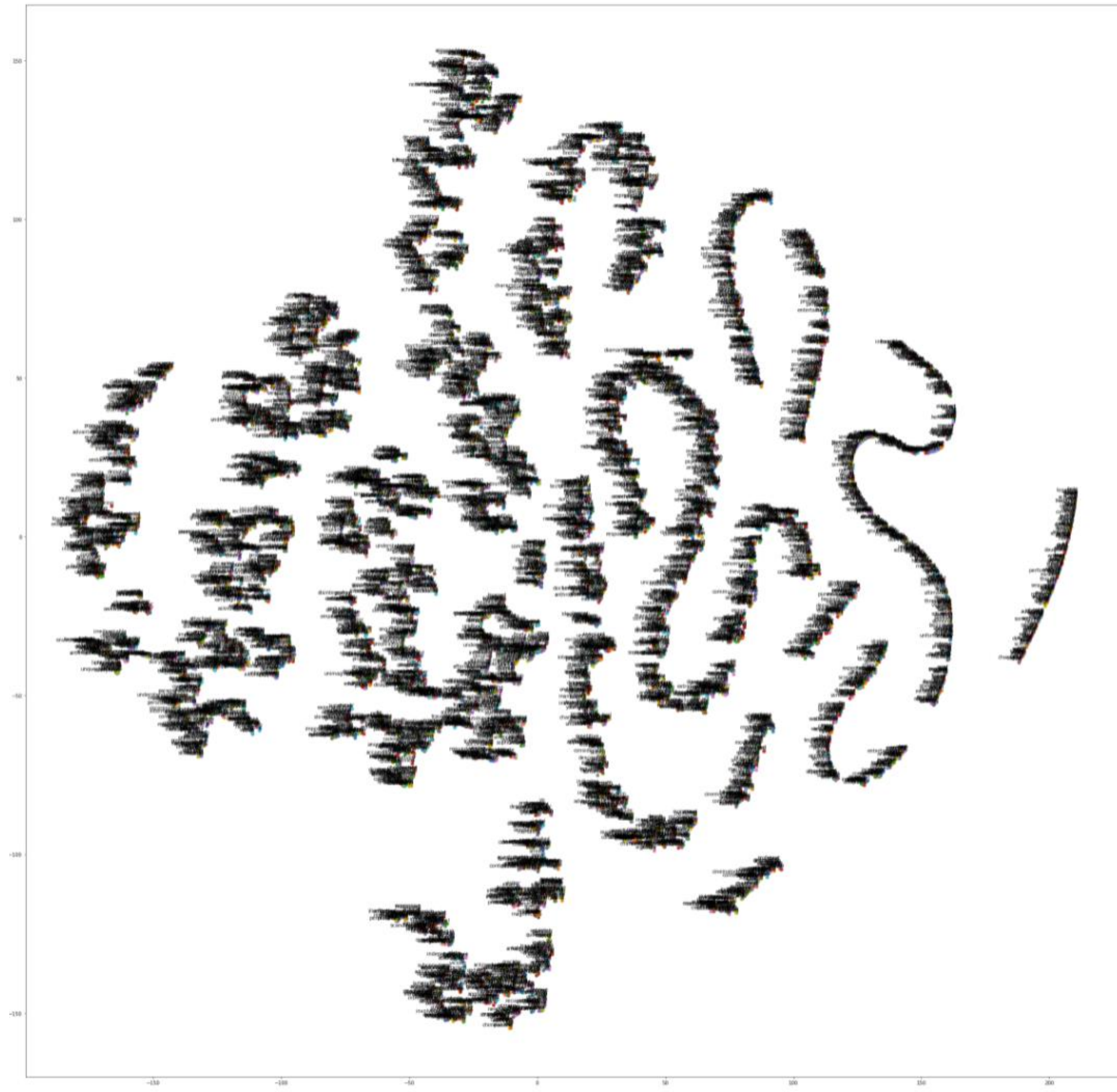
T-SNE words2vec

Vectorized Representation



Vectorized Representation

T-SNE Word2Vec

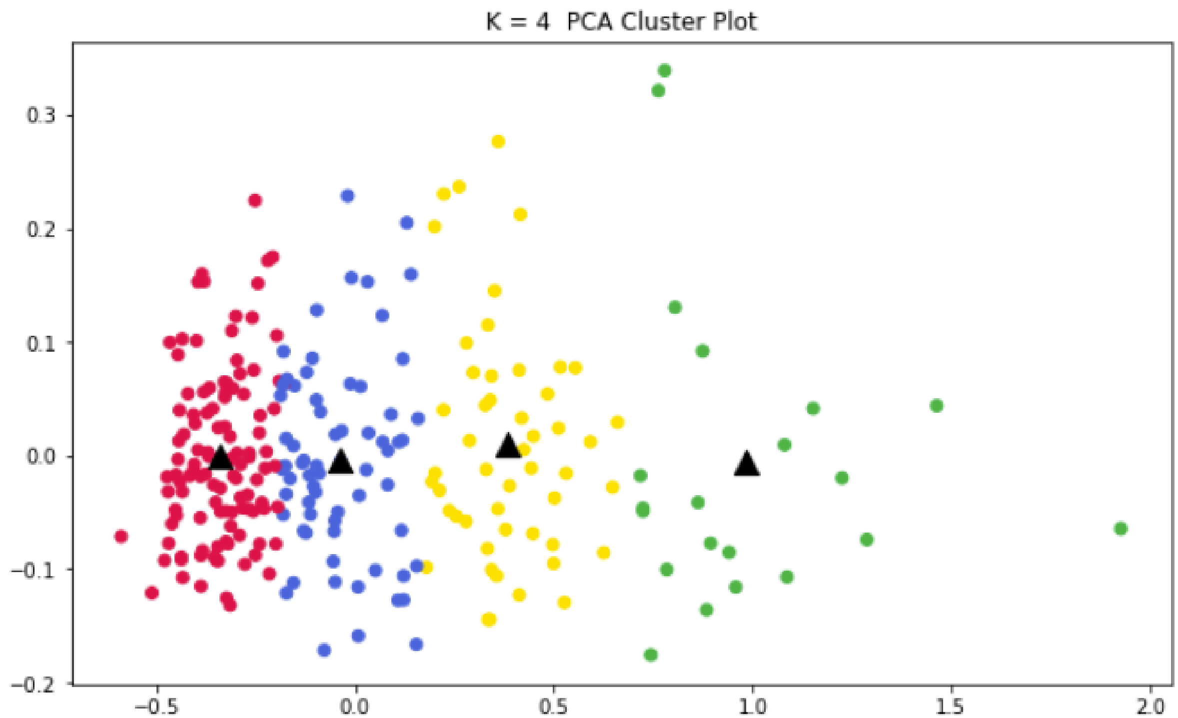


Appendix B

Method 1 Word2Vec Doc2Vec Embedding size 100

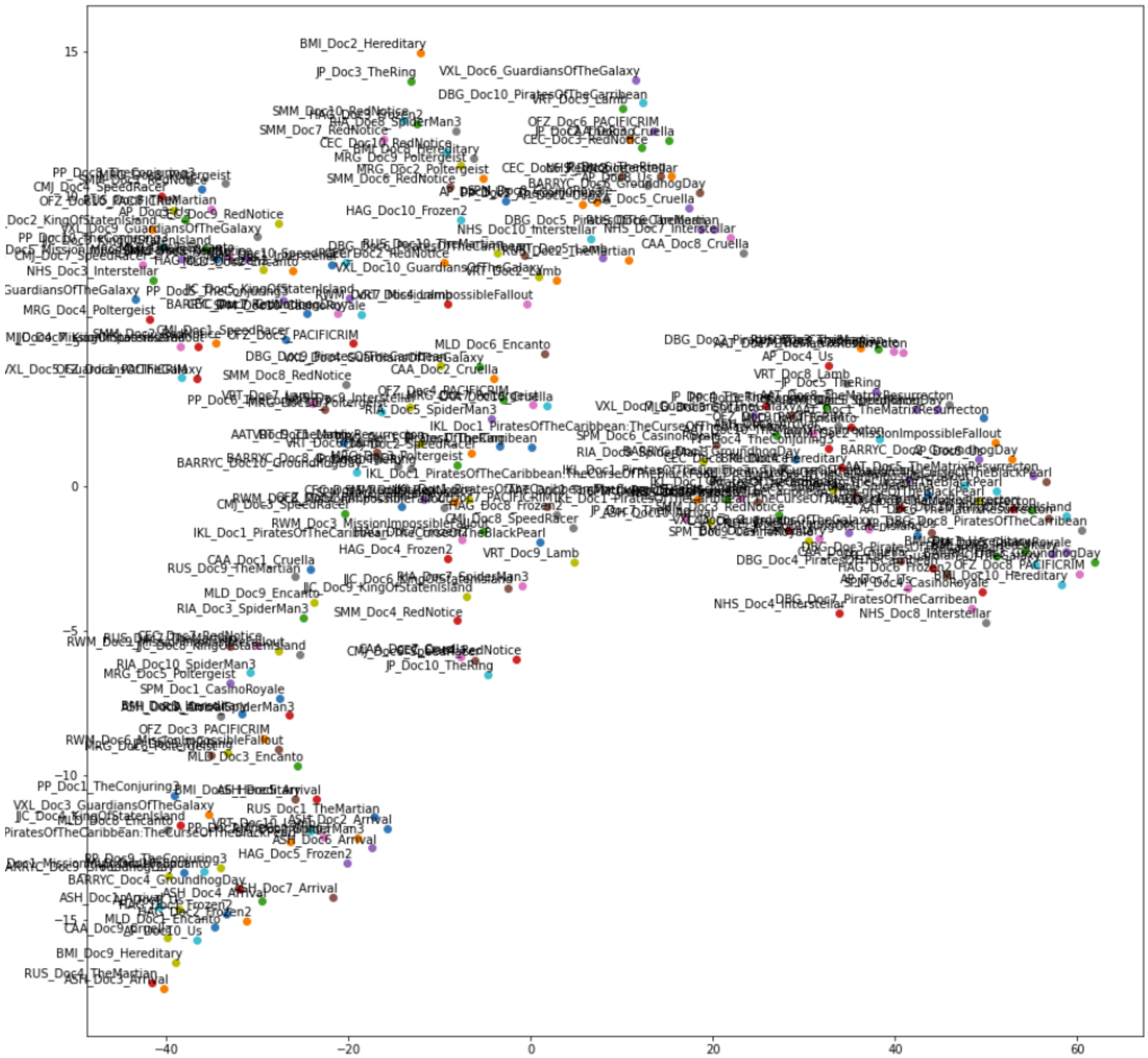
Vectorized Representation

K-means dov2vec



Vectorized Representation

T-SNE doc2vec



Vectorized Representation

T-SNE word2vec