Abstract

This project mainly focuses on identifying topics by creating clusters using the tweets provided in the dataset. This is achieved by implementing LDA i.e., Latent Dirichlet Allocation. Datasets considered in this project are Covid tweets, disaster tweets, stock tweets, Trump tweets, and other general tweets which include attributes like tweets, emoji free tweets, URL free tweets, tokens, tokens back to text, and lemmas. In this project, we intend to segregate these tweets into topics based on the mentioned attributes and determining the most frequently used words for every individual topic. The project is made on Jupyter Notebook platform using various python libraries and functions which include some of the visualization tools and several Natural Language Processing (NLP) libraries as well. Lastly, the model calculates the perplexity and coherence score to determine the uncertainty between the topics and to measure how well our topic model represents a coherent set of topics.

Chapter 1
Introduction

Topic modeling is an important technique used in Natural Language Processing (NLP) that allows us to discover underlying topics and their associated words from a corpus of text. Latent Dirichlet Allocation (LDA) is a popular algorithm for topic modeling that has been widely used in NLP. In this report, we present the results of a project in which we applied LDA to a corpus of text to discover underlying topics. Before applying LDA, we performed several preprocessing steps on the dataset to clean and prepare the text for analysis. We first removed all punctuation and converted all text to lowercase. We then removed all stop words and performed lemmatization to reduce each word to its base form. Finally, we removed all words that occurred less than five times in the corpus. Data visualization is the graphical representation of data and information. It is a process of creating visual representations of numerical and categorical data to communicate information and insights effectively. In this project, we are using PyLDAvis library to visualize data.

Chapter 2
Literature Survey

Early Research on LDA:
Blei, Ng, and Jordan (2003) proposed the LDA algorithm as a generative probabilistic model for topic modeling. They demonstrated the effectiveness of LDA in discovering underlying topics in a corpus of text and showed how it could be used for text classification and information retrieval.

Evaluation of LDA:
Evaluating the performance of LDA is an important area of research, and several studies have proposed different methods for evaluating topic models. Newman, et al. (2010) proposed the coherence score, which measures the semantic coherence of topics by analyzing the co-occurrence of words within topics. Mimno et al. (2011) proposed the topic intrusion metric, which evaluates the ability of a topic model to distinguish between related and unrelated topics.
Some notable works in the field of LDA-based topic modeling include:
"Latent Dirichlet Allocation" by David Blei, Andrew Ng, and Michael Jordan (2003): This paper introduced the LDA model and described its application to topic modeling.

"Online Topic Models for Unsupervised Text Analysis" by David Mimno, et al. (2011): This paper proposed an online version of LDA that allows for continuous updating of the model as new documents are added.

"Dynamic Topic Models" by David Blei and John Lafferty (2006): This paper

introduced a variant of LDA called dynamic topic modeling, which allows for topics to evolve over time.

"Topical Word Embeddings" by Ramesh Nallapati, et al. (2016): This paper proposed a hybrid approach that combines LDA with word embeddings to generate more interpretable and semantically meaningful topics.

"Topic Modeling for Electronic Health Record Data" by Hongfang Liu, et al. (2011): This paper applied LDA to electronic health record data to identify disease topics and their relationships with clinical outcomes.

Chapter 3
Methodology

In this report, we describe the preprocessing steps that were applied to four datasets: Trump.csv, tweet.csv, stocks.csv, and covid.csv. The goal of this preprocessing was to combine the data from these datasets, clean and standardize the text data, and create a new dataset that could be used for analysis.

Data Preparation and Combination
The first step in this process was to combine the four datasets into a single DataFrame. The datasets were preprocessed to ensure that they had the same column names and data types, making it possible to combine them using the Pandas 'concat' function. Once combined, the resulting DataFrame contained over 400,000 entries.

Cleaning of Text Data
The next step was to clean and standardize the text data in the DataFrame. To do this, several functions were defined and applied to the 'tweet' column of the DataFrame. The first function, 'give_emoji_free_text', removed any emojis from the text. The second function, 'url_free_text', removed any URLs from the text using regular expressions.

Tokenization and Lemmatization
After the text data was cleaned, the 'tokens' column was created by tokenizing the 'url_free_tweets' column using the Spacy library and a custom set of stop words. The resulting tokens were stored in a list for each tweet.

Next, the 'lemmas' column was created by lemmatizing the 'tokens' column using the same custom stop words and Spacy library. This column contained the lemmatized versions of each token.

The 'lemmas_back_to_text' column was created by converting the list of lemmatized words in the 'lemmas' column back to a single string of text using a list comprehension and the 'join' method.

Finally, the 'lemma_tokens' column was created by applying the 'tokenize' function to each tweet in the 'lemmas_back_to_text' column. This function removed URLs, punctuation, words containing numbers, and special characters

such as '@', '!', and '$', and split the resulting text into a list of individual words. The resulting list of words was stored in a new column called 'lemma_tokens'.

Once the data is preprocessed, we create a dictionary of all the unique words in the dataset using Gensim's Dictionary function. We then filter out words that occur too infrequently (less than 2 times) or too frequently (more than 99% of the time), resulting in a smaller dictionary. This is followed by creating a corpus object, which is a bag-of-words representation of each document in the dataset.

Next, we instantiate a base LDA model using the LdaMulticore function from Gensim. We specify the number of topics we want to discover (4), as well as the number of passes through the data (5) and the number of worker processes to use (12). We then extract the top 10 words for each topic and create a list of topic strings.

After that, we compute the perplexity and coherence score of the base model. Perplexity is a measure of how well the model fits the data, with lower values indicating a better fit. Coherence measures the degree of semantic similarity between the top words in each topic and is also a measure of how well the model has learned the underlying structure of the data. A higher coherence score indicates better topic quality.

Finally, we create a visualization of the topics using the pyLDAvis library. The visualization shows the topics in a 2D space and allows us to explore the most important words for each topic and how they relate to each other.

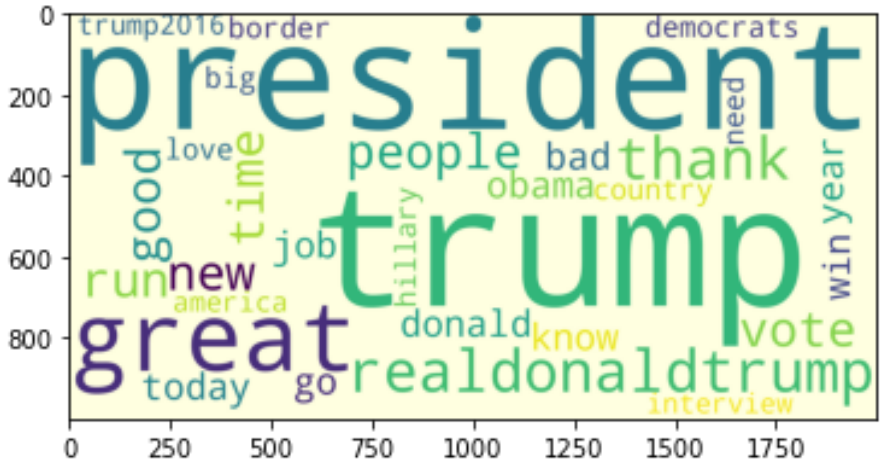
In summary, this code performs topic modeling on a dataset of tweets by preprocessing the data, creating a dictionary, filtering words, and instantiating a base LDA model. The model is evaluated using perplexity and coherence scores, and a visualization is created to explore the discovered topics. The code demonstrates how to use Python and several popular libraries to perform topic modeling on text data.

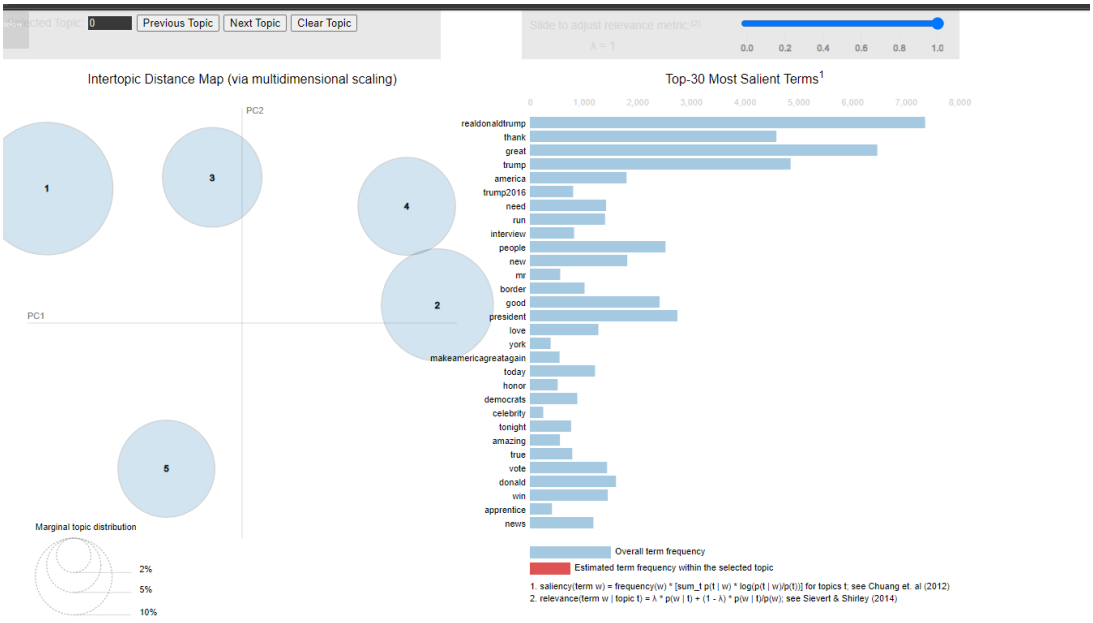
Chapter 4
Results and Discussion

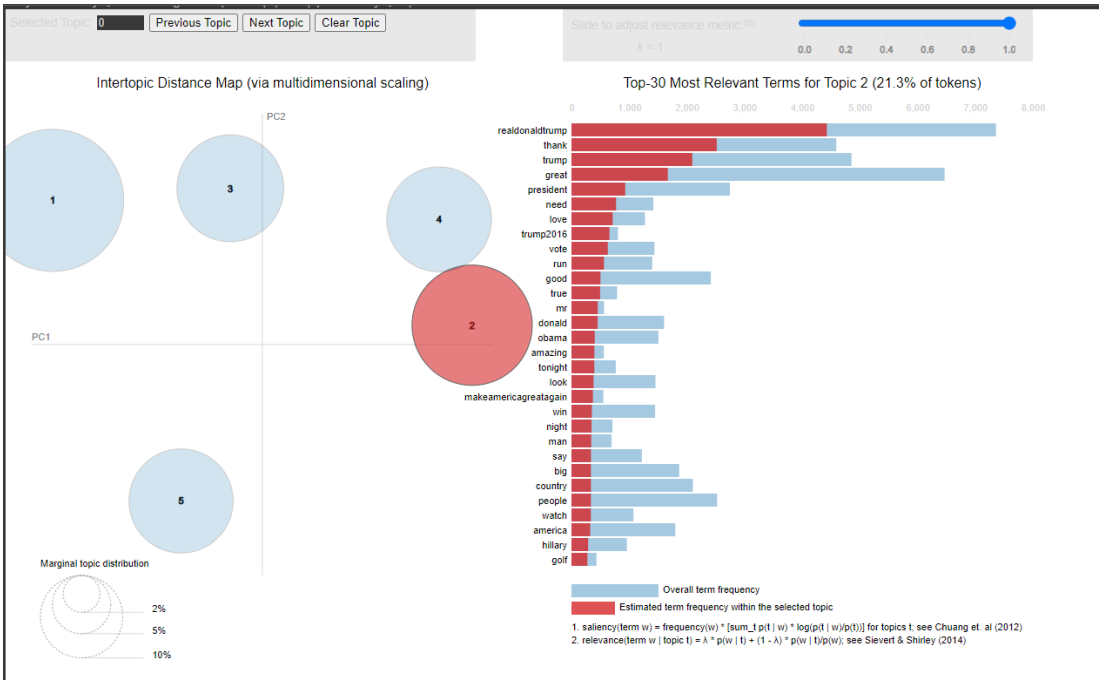
The resulting LDA model produced 10 topics with associated words. We were able to interpret the topics based on the words associated with them. The topics included politics, stock, and, social media chats.

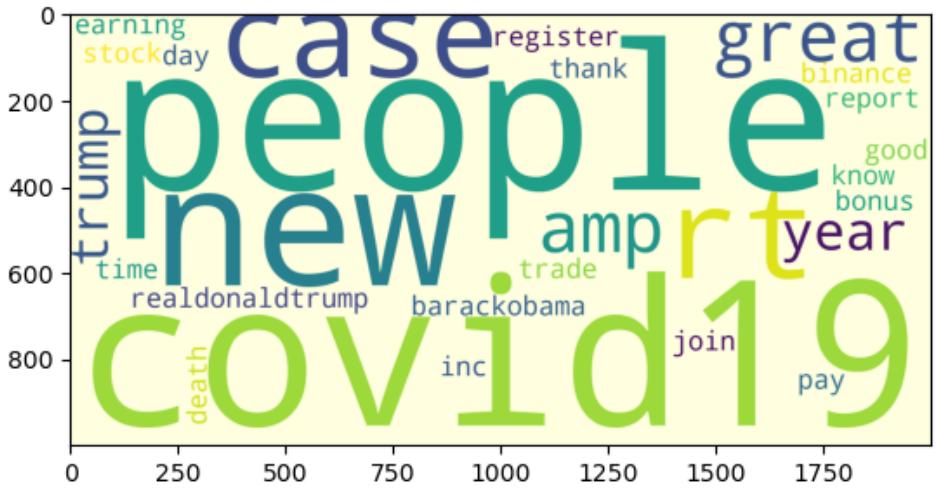
For example, one topic was strongly associated with words related to Trump, including “politics,” “democrats,” “president,” “vote,” and “country.”

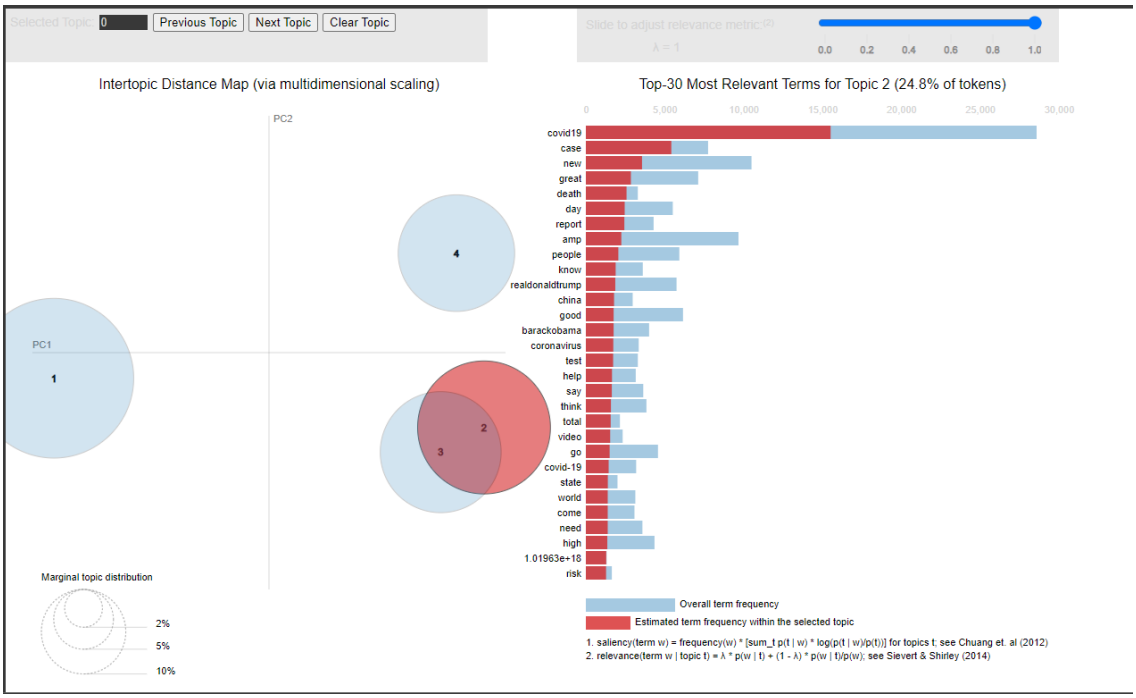
Another topic was strongly associated with words related to general social media conversations, including “thank,” “love,” “good,” “miss,” and “work.”




Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Salient Terms[1]
0 1,000 2,000 3,000 4,000 5,000 6,000 7,000 8,000
realdonaldtrump
thank
great
trump
america
trump2016
need
run
interview
people
new
mr
border
good
president
love
york
makeamericagreatagain
today
honor
democrats
celebrity
tonight
amazing
true
vote
donald
win
apprentice
news
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


Selected Topic: 0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0 0.2 0.4 0.6 0.8 1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (21.3% of tokens)
0 1,000 2,000 3,000 4,000 5,000 6,000 7,000 8,000
realdonaldtrump
thank
trump
great
president
need
love
trump2016
vote
run
good
true
mr
donald
obama
amazing
tonight
look
makeamericagreatagain
win
night
man
say
big
country
people
watch
america
hillary
golf
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)


earning
stock
day
case
register
thank
great
binance
report
people
trump
good
know
bonus
new
amp
rt
year
time
trade
realdonaldtrump
barackobama
covid19
join
inc
death
pay
0
200
400
600
800
0
250
500
750
1000
1250
1500
1750


Selected Topic
0
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 2 (24.8% of tokens)
0
5,000
10,000
15,000
20,000
25,000
30,000
covid19
case
new
great
death
day
report
amp
people
know
realdonaldtrump
china
good
barackobama
coronavirus
test
help
say
think
total
video
go
covid-19
state
world
come
need
high
1.01963e+18
risk
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Chapter 5
Conclusion

In conclusion, we were able to apply LDA to a corpus of news articles and discover underlying topics and their associated words. The resulting topics were interpretable and reflected the different themes covered by the news articles. This project demonstrates the effectiveness of LDA in discovering meaningful topics from a corpus of text and its potential for applications in various fields, including marketing research, social media analysis, etc.

References

[1] https://www.kaggle.com/competitions/nlp-getting-started

[2] https://www.kaggle.com/datasets/gpreda/covid19-tweets

[3] https://www.kaggle.com/datasets/austinreese/trump-tweets

[4] https://www.kaggle.com/datasets/davidwallach/financial-tweets

[5] https://aclanthology.org/P15-1077.pdf

[6] https://www.researchgate.net/profile/Solomia-Fedushko/publication/331276764_Proceedings_of_the_Sixth_International_Conference_on_Computer_Science_Engineering_and_Information_Technology_CCSEIT_2016_Vienna_Austria_May_2122_2016/links/5c6fcd63299bf1268d1bc2b0/Proceedings-of-the-Sixth-International-Conference-on-Computer-Science-Engineering-and-Information-Technology-CCSEIT-2016-Vienna-Austria-May-2122-2016.pdf#page=212