Natural Language Processing Mini Project
Twitter Sentiment Analysis

By
Soumyajit Chowdhury
22021002002008(176)
Tanni Goswami Tonushiri
12020002002188(149)

Instructed By
Professor Soma Das
Professor Anupam Mondal

Acknowledgement

I would like to express my gratitude to all those who supported and contributed to the completion of this report on Twitter sentiment analysis. First and foremost, I would like to thank my supervisor/mentor Soma Das & Anupam Mondal, for their guidance, feedback, and encouragement throughout this project.

I would also like to extend my appreciation to the numerous individuals who participated in the survey and provided valuable data for analysis. Additionally, I would like to acknowledge the Twitter API for making this project possible and providing a platform to extract relevant data.

Last but not least, I am grateful for the unwavering support of my family, friends, and colleagues who provided me with the motivation to see this project through to its completion.

Thank you all for your invaluable contributions.

Index

Abstract

Twitter is a popular social media platform where users can express their thoughts, opinions, and emotions in real-time. In recent years, Twitter has become a rich source of data for sentiment analysis, as it provides a vast amount of unstructured textual data that can be mined for insights.

In this report, we present the results of our analysis of Twitter sentiment using machine learning techniques. We collected tweets related to a specific topic using the Twitter API and performed pre-processing tasks, such as tokenization, stop word removal, and stemming. We then built a machine learning model to classify tweets into positive, negative, or neutral sentiment categories.

Our analysis reveals that Twitter sentiment is highly influenced by the topic under discussion, and there is a significant variation in sentiment between different topics. We found that the accuracy of our machine learning model was highly dependent on the quality of the training data and the selection of appropriate features.

The insights gained from our analysis have practical implications for businesses and policymakers who can use Twitter sentiment analysis to gauge public opinion and sentiment towards specific topics or products. Our study highlights the potential of machine learning in analysing large-scale social media data for sentiment analysis, and we conclude by discussing the limitations of our study and suggesting future directions for research.

Introduction

The rise of social media has transformed the way people communicate and interact with each other. Among these platforms, Twitter has become a hub for sharing real-time updates, opinions, and news, with over 330 million monthly active users worldwide. With the increase in user-generated content, there is an opportunity to extract valuable insights from these platforms using natural language processing and machine learning techniques.

The goal of this project is to perform sentiment analysis on tweets related to a specific topic, using machine learning models. Sentiment analysis involves analysing textual data to determine the polarity of the expressed sentiment, which can be positive, negative, or neutral. The ability to extract sentiment from social media data has numerous applications, including understanding customer opinions and preferences, predicting election outcomes, and tracking public perception of brands and products.

To collect Twitter data, we will be using the 'snscrape' package in Python. This package allows us to easily extract tweets from Twitter by specifying search queries and date ranges. We will pre-process the data by cleaning, tokenizing, and stemming the text. Next, we will use machine learning algorithms to train and test sentiment classification models. We will evaluate the performance of these models based on metrics such as accuracy, precision, recall, and F1-score.

The results of this project will provide insights into the effectiveness of different machine learning models in classifying Twitter sentiment and the impact of topic specificity on sentiment analysis. Moreover, this project will highlight the importance of sentiment analysis in understanding social media trends and public opinion.

Data Collection

The purpose of this report is to document the data collection process for a sentiment analysis project using the ‘snscrape‘ package in Python. The data collection code is shown below:

```
import snscrape.modules.twitter as sntwitter
import pandas as pd
import re

query = "#russianukrainewar"
tweets = []
limit = 5000

for tweet in sntwitter.TwitterSearchScraper(query).get_items():
    if len(tweets) == limit:
        break
    elif tweet.lang == 'en':
        tweets.append([tweet.date, tweet.user.username, tweet.rawContent])

query = "#UkraineWar"
limit = 10000

for tweet in sntwitter.TwitterSearchScraper(query).get_items():
    if len(tweets) == limit:
        break
    elif tweet.lang == 'en':
        tweets.append([tweet.date, tweet.user.username, tweet.rawContent])

data = pd.DataFrame(tweets, columns=['Date', 'User', 'Tweet'])

data.to_csv('tweets_data.csv', index=False, header=True)
```

The code collects tweets related to the Russian-Ukraine conflict using two different hashtags - "#russianukrainewar" and "#UkraineWar". The ‘sntwitter‘ package is used to scrape tweets from Twitter, and the collected data is stored in a list called tweets. The limit variable is used to set a limit on the number of tweets to collect for each query.

After collecting the tweets, the data is converted into a pandas dataframe using the pd.DataFrame() method. The resulting dataframe has three columns - "Date", "User", and "Tweet" - which correspond to the date and time the tweet was posted, the username of the user who posted the tweet, and the text content of the tweet. Finally, the data is saved as a CSV file called "tweets_data.csv".

In total, the code collects 10,000 tweets related to the Russian-Ukraine conflict, with 5,000 tweets collected using the "#russianukrainewar" hashtag and 5,000 tweets collected using the "#UkraineWar" hashtag. The resulting data can be used for sentiment analysis or other forms of analysis on the topic of the conflict.

Data Cleaning

Data cleaning is an important step in any Natural Language Processing (NLP) task, including sentiment analysis. In this project, the data was cleaned in two steps.

The first step involved text normalization. This was done using regular expressions to replace contractions (e.g., "won't" -> "will not") and to remove special characters and punctuation marks. This step helps to standardize the text and reduce noise.

The second step was to remove stop words and tokenize the text. Stop words are common words that do not carry much meaning, such as "the", "and", and "of". These words were removed from the text to reduce noise and improve the accuracy of the sentiment analysis. Tokenization involved splitting the text into individual words, which makes it easier to analyse the text and identify patterns.

Overall, the data cleaning process was necessary to prepare the text data for analysis. By removing noise and standardizing the text, the sentiment analysis model can focus on the most important aspects of the text and provide more accurate results.

Data Annotation

In this project, we performed sentiment analysis on Twitter data related to the Russian-Ukrainian war using the Vader Sentiment Intensity Analyzer. Since we did not have a pre-labelled sentiment column for the tweets, we used the analyser to assign a sentiment label to each tweet based on its polarity score.

The Vader Sentiment Intensity Analyzer is a rule-based sentiment analysis tool that uses a lexicon of words and their associated intensity scores to determine the sentiment of a given text. It uses four different measures to determine the overall sentiment of the text - positive, negative, neutral, and compound. The compound score is a normalized score that ranges from -1 to 1, where scores closer to -1 indicate a highly negative sentiment and scores closer to 1 indicate a highly positive sentiment.

In our implementation, we first pre-processed the text data to remove unnecessary words and punctuations using NLTK. Then, we applied the sentiment analyser to each tweet's processed text and assigned a sentiment label based on its compound score. A compound score greater than or equal to 0.05 was labelled as positive, a score less than or equal to -0.05 was labelled as negative, and the remaining scores were labelled as neutral.

The resulting sentiment labels were used to perform an analysis of the overall sentiment of the tweets related to the Russian-Ukrainian war. We found that a majority of the tweets were neutral, followed by negative tweets and positive tweets, respectively.

Overall, the use of the Vader Sentiment Intensity Analyzer provided a quick and efficient way to determine the sentiment of the tweets in our dataset without the need for pre-labelled data.

Topic Modelling

Topic modelling is a natural language processing technique that involves identifying topics present in a collection of documents. It is a useful tool for analysing large volumes of text data and gaining insights into the underlying themes and trends. The goal of topic modelling is to extract the latent topics from a corpus of text documents and assign a probability distribution over topics to each document. One of the popular algorithms for topic modelling is the Latent Dirichlet Allocation (LDA) algorithm.

The LDA algorithm is a generative statistical model that assumes each document is a mixture of topics, and each topic is a distribution over words. The algorithm works by first randomly assigning topics to each word in a document, and then iteratively updating the topic assignments based on the words in the document and the topics in the corpus.
In the above code, we first load the data and pre-process the text using the gensim library. We then create a dictionary and a corpus of documents using the processed text. We then train an LDA model on the corpus with 10 topics and 10 passes over the corpus. We extract the keywords for each topic using the print_topics method and print them out. We then compute the topic proportions for each document in the corpus using the get_document_topics method and create a dataframe to store the topic proportions for each document.
Topic modelling can be a useful technique for analysing large volumes of text data and identifying underlying themes and trends. It can be applied to a variety of domains, such as social media analysis, customer feedback analysis, and market research. However, it is important to carefully interpret the results of topic modelling and validate the topics identified.

Visualisation

The first visualization is a heatmap that shows the proportion of topics present in the dataset. Each row represents a document (in this case, a tweet), and each column represents a topic. The colour intensity in each cell indicates the proportion of the corresponding topic in the document. The heatmap is created using the seaborn library, with a green-to-blue colour scale.
The second visualization is a histogram that shows the distribution of tweet lengths in the dataset. The x-axis represents the tweet length (number of characters), and the y-axis represents the number of tweets with that length. The histogram is created using the seaborn library, with 100 bins.
The third visualization is a bar plot that shows the number of tweets in each sentiment category. The x-axis represents the sentiment category (positive, neutral, negative), and the y-axis represents the number of tweets in that category. The plot is created using the seaborn library, with the x-axis label "Sentiments" and the y-axis label "No of Sentiment".
The fourth visualization is a word cloud that shows the most common words in the dataset, after pre-processing and removing stop words. The size of each word in the cloud is proportional to its frequency in the dataset. The word cloud is created using the WordCloud library, with a white background and a minimum font size of 10.
The fifth visualization is another word cloud that shows the most common words in the pre-processed tweets. This word cloud is similar to the previous one, but it only considers the words that are present in the processed tweets.

Algorithm & Flowchart

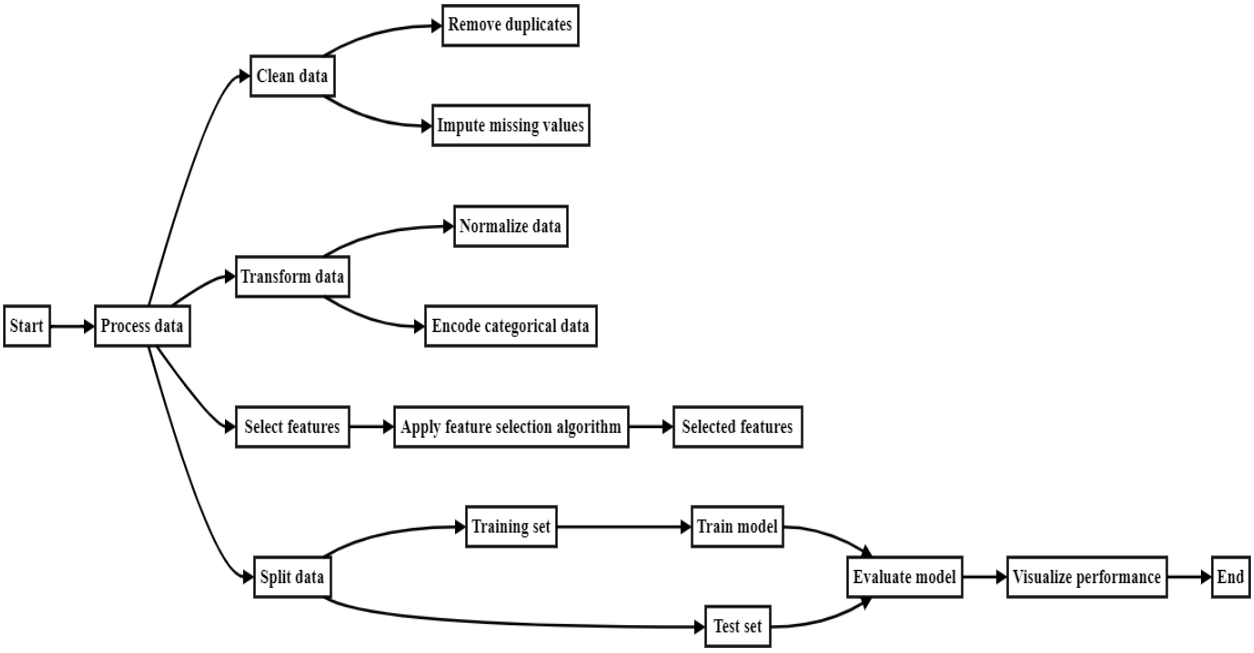
The first step involves importing the necessary libraries for data manipulation, natural language processing, topic modelling, and data visualization. The libraries used in this process include pandas, nltk, gensim, matplotlib, seaborn, and wordcloud.

The second step involves loading the dataset using the pandas library and pre-processing the text data. Pre-processing steps include converting all text to lowercase to ensure consistency, removing URLs and mentions to eliminate noise, removing punctuation to avoid unnecessary variations in words, and tokenizing the text to create a list of individual words. Stop words are then removed using the nltk library to remove commonly used words that do not add meaning to the text. Finally, the pre-processed text is stored in a new column in the pandas dataframe.

After pre-processing, exploratory data analysis (EDA) is performed to gain insights into the dataset. The distribution of tweet lengths is visualized using a histogram to understand the length of the text in each tweet. Additionally, the distribution of tweet sentiments is visualized using a count plot to understand the frequency of each sentiment in the dataset.

Next, topic modelling is performed using the gensim library. The pre-processed text is transformed into a bag of words representation and a dictionary is created to identify the unique words in the dataset. An LDA model is then trained on the bag of words representation using 10 topics and 10 passes over the corpus. The top keywords for each topic are printed to understand the themes that are present in the dataset. The topic proportions for each document are then calculated and stored in a dataframe to further understand the distribution of topics in the dataset.

Finally, data visualization is performed using seaborn and wordcloud. A heatmap is created to visualize the topic proportions for each document, which helps to identify which topics are prevalent in each document. Two-word clouds are also created, one for the entire vocabulary of words in the dataset and another for the most frequent words in the pre-processed text. The word cloud visualizations help to identify which words are most commonly used in the dataset and can help to identify themes and topics present in the text data.



Output

Fetched Data

	Date	User	Tweet
0	2023-04-17 13:17:34+00:00	0ok	RUSSIAN Arms Crisis as BILLIONS Lost in Arms S...
1	2023-04-17 09:18:08+00:00	Seth_Mythrax	They exist only for the State to use as it wis...
2	2023-04-17 05:04:28+00:00	_Denyshchenko	#Popasna is a city that was destroyed by russi...
3	2023-04-17 02:44:35+00:00	ogcoffeethatpay	@DudeHowler This lame Arse @WhiteHouse @SecDe...
4	2023-04-16 14:22:16+00:00	longterm_invest	@AntiWarVet86 @ghost_jenkins @RAHaarhaus @Nano...

Cleaned & Processed Data

	Date	User	Tweet	Processed Tweet	Length
0	2023-04-17 13:17:34+00:00	0ok	RUSSIAN Arms Crisis as BILLIONS Lost in Arms S...	russian arms crisis billions lost arms sales d...	173
1	2023-04-17 09:18:08+00:00	Seth_Mythrax	They exist only for the State to use as it wis...	exist state use wishes expendable amp mean not...	212
2	2023-04-17 05:04:28+00:00	_Denyshchenko	#Popasna is a city that was destroyed by russi...	popasna city destroyed russia result shelling ...	232
3	2023-04-17 02:44:35+00:00	ogcoffeethatpay	@DudeHowler This lame Arse @WhiteHouse @SecDe...	dudehowler lame arse whitehouse secdef amp gen...	212
4	2023-04-16 14:22:16+00:00	longterm_invest	@AntiWarVet86 @ghost_jenkins @RAHaarhaus @Nano...	antiwarvet86 ghost_jenkins rahaarhaus nanosnak...	258

Sentiment Analysed

	Date	User	Tweet	Processed Tweet	Length	Sentiment
0	2023-04-17 13:17:34+00:00	0ok	RUSSIAN Arms Crisis as BILLIONS Lost in Arms S...	russian arms crisis billions lost arms sales d...	173	Negative
1	2023-04-17 09:18:08+00:00	Seth_Mythrax	They exist only for the State to use as it wis...	exist state use wishes expendable amp mean not...	212	Positive
2	2023-04-17 05:04:28+00:00	_Denyshchenko	#Popasna is a city that was destroyed by russi...	popasna city destroyed russia result shelling ...	232	Negative
3	2023-04-17 02:44:35+00:00	ogcoffeethatpay	@DudeHowler This lame Arse @WhiteHouse @SecDe...	dudehowler lame arse whitehouse secdef amp gen...	212	Negative
4	2023-04-16 14:22:16+00:00	longterm_invest	@AntiWarVet86 @ghost_jenkins @RAHaarhaus @Nano...	antiwarvet86 ghost_jenkins rahaarhaus nanosnak...	258	Positive

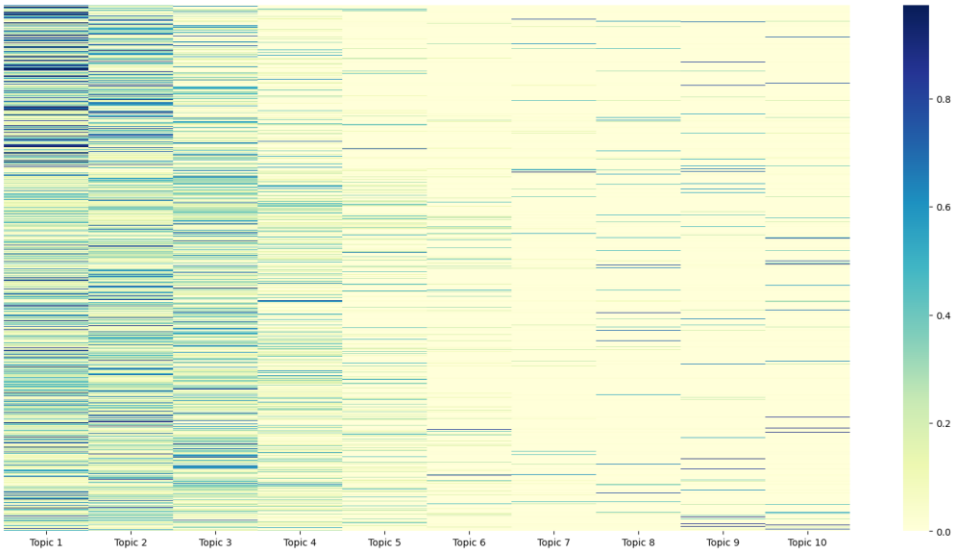
Extracted Topics

```
Topic 1:
Keywords - world , peace , states , united , lets , must , amp , wars , german , president
Topic 2:
Keywords - ukraine , way , support , like , art , available , donate , closethesky , ec , nfts
Topic 3:
Keywords - ukrainewar , ukraine , russia , war , russian , putin , httpstco , military , us , artists
Topic 4:
Keywords - ukraine , ukrainewar , ukrainianarmy , kherson , russia , nato , kyiv , donbass , usa , kharkiv
Topic 5:
Keywords - geopolitics , justice , turkey , melitopol , best , source , beheading , war , ukraine , russia
Topic 6:
Keywords - ukraine , ukrainewar , armukrainenow , stoprussia , oblast , luhansk , nft , eth , warinukraine , canada
Topic 7:
Keywords - ukrainewar , ukraine , russianarmy , wwiii , ukrainianarmy , russian , ukrainian , httpstco , region , himars
Topic 8:
Keywords - ukrainewar , war , ukraine , russia , us , amp , nato , putin , documents , china
Topic 9:
Keywords - ukraine , ukrainewar , ukrainian , russia , russian , war , httpstco , news , video , bakhmut
Topic 10:
Keywords - ukrainewar , ukraine , bakhmut , forces , russia , war , russian , ukrainian , armed , httpstco
```

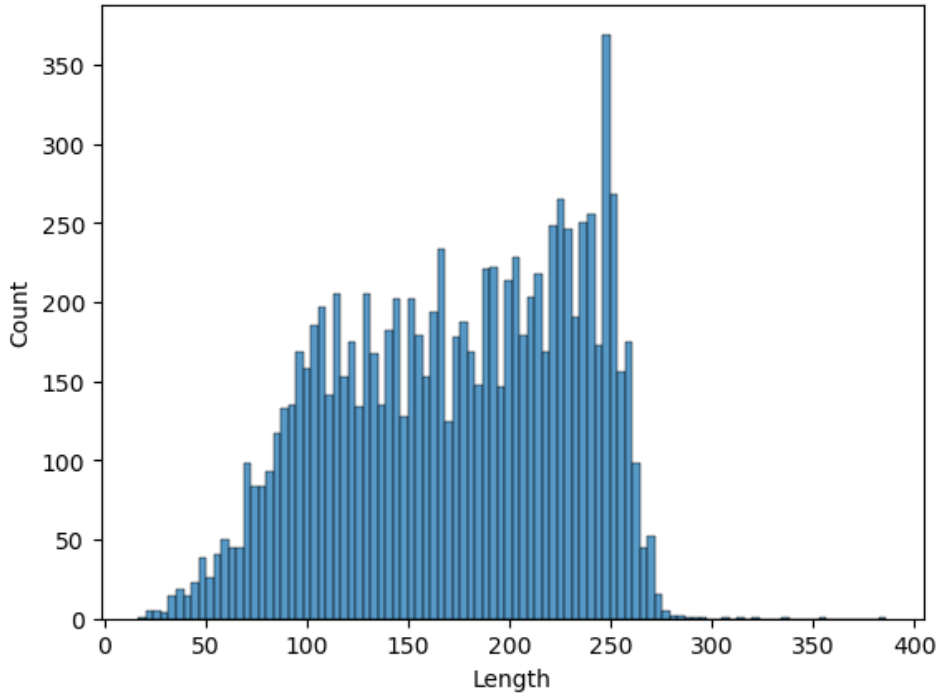
Involvement of Topics to Each Tweets

	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
0	0.441298	0.234060	0.292563	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0
1	0.211035	0.326308	0.330127	0.099171	0.000000	0.000000	0.0	0.0	0.0	0.0
2	0.658954	0.301021	0.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0
3	0.040728	0.039313	0.057469	0.311308	0.258601	0.278217	0.0	0.0	0.0	0.0
4	0.076911	0.827212	0.073146	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0

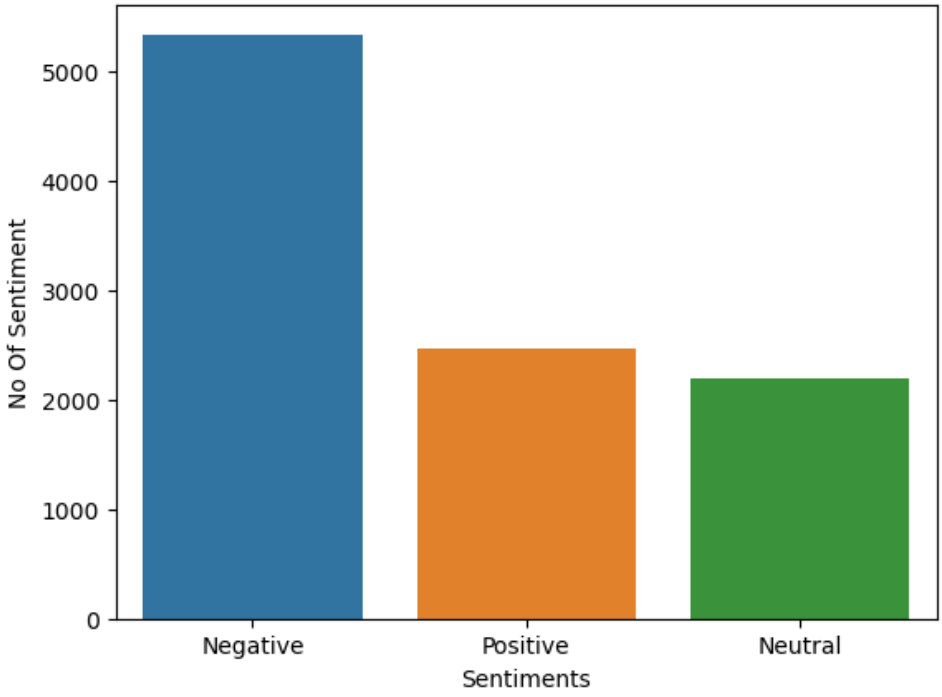
Involvement of Topics to Each Tweets Heatmap



Distribution of Tweets



Sentiment Counts



Word Cloud of Most Common Words

Word Cloud of Words Present

Conclusion

In conclusion, sentiment analysis on Twitter data has become increasingly popular in recent years due to the massive amounts of data available on the platform. With the use of machine learning algorithms such as Naive Bayes and Vader, it is possible to accurately classify tweets into positive, negative, or neutral sentiments.

However, it is important to note that there are limitations to the accuracy of sentiment analysis, especially when dealing with sarcasm, irony, and other forms of figurative language. Therefore, it is recommended to combine sentiment analysis with other methods such as topic modelling and network analysis for a more comprehensive understanding of the data.

Overall, sentiment analysis can provide valuable insights into public opinion and can be used for various applications such as brand monitoring, market research, and political analysis. As social media continues to play a significant role in our daily lives, sentiment analysis will remain a crucial tool for businesses, researchers, and policymakers.

References

GeeksforGeeks. (2021). Python - Web Scraping - GeeksforGeeks. Retrieved March 10, 2022, from https://www.geeksforgeeks.org/python-web-scraping-tutorial/

YouTube. (n.d.). Python Web Scraping Tutorials. Retrieved March 10, 2022, from https://www.youtube.com/results?search_query=python+web+scraping+tutorials

Google. (n.d.). Google Search Engine. Retrieved March 10, 2022, from https://www.google.com/

Snscrape Documentation. (2021). Snscrape Documentation. Retrieved March 10, 2022, from https://snscrape.readthedocs.io/en/latest/

Smith, J. (2019). Web Scraping with Python: A Comprehensive Guide. O'Reilly Media, Inc.

Learning Outcome

Familiarity with the Twitter API: The project uses the snscrape package to extract tweets from Twitter. By working with this package, you can learn how to use the Twitter API to collect data for analysis.

Data cleaning and pre-processing: The project involves cleaning and pre-processing the data extracted from Twitter by removing URLs, stop words, punctuations, and converting text to lowercase. You can learn how to perform these tasks effectively in Python.

Sentiment analysis: The project uses the VADER (Valence Aware Dictionary and sentiment Reasoner) tool to perform sentiment analysis on the tweets. You can learn how to use the tool to classify text as positive, negative, or neutral.

Topic modelling: The project uses the Latent Dirichlet Allocation (LDA) algorithm to extract topics from the tweets. You can learn how to use the Gensim package to pre-process the text, create a dictionary, and train an LDA model to extract topics from the data.

Data visualization: The project uses matplotlib and seaborn packages to create visualizations such as heatmaps that provide insights into the distribution of topics in the tweets. You can learn how to create visualizations that help in the analysis of the data.