Topic Modeling Using Twitter and LDA

Mohammed Abdul Basith
Artificial intelligence and machine learning
Lambton college
Toronto, Canada
thatsbasith@gmail.com

Mohammed Salman Khan
Artificial intelligence and machine learning
Lambton college
Toronto, Canada
mdnawaz931@gmail.com

Tasneem Badar
Artificial intelligence and machine learning
Lambton college
Toronto, Canada
tasneembadar123@gmail.com

Saniya Jabeen
Artificial intelligence and machine learning
Lambton college
Toronto, Canada
saniyajabeen17@gmail.com

***Abstract*—In this project, we explore the application of Latent Dirichlet Allocation (LDA) for topic modeling on Twitter data. The objective is to develop Python scripts capable of downloading Tweets from specific Twitter accounts or using any News API, performing data preprocessing, applying LDA to extract latent topics, generating word clouds based on topic-term distributions, and enabling interactive exploration of related tweets. The methodology involves four main steps: first, data collection and cleaning; second, LDA modeling to identify 10-15 topics from the corpus; third, visualization of topic-term associations using word clouds; and finally, linking words in the word cloud to relevant tweets, enhancing user interaction and understanding of the generated topics. The project aims to provide a practical tool for exploring thematic trends in social media or news articles, demonstrating the utility of LDA in extracting meaningful insights from unstructured textual data.**

Keywords— Topic modeling, Latent Dirichlet Allocation (LDA), Twitter data, Python, data preprocessing, word clouds, interactive visualization, thematic trends, social media analysis, news analysis

I. Introduction

In the realm of social media and online news, the vast amount of textual data generated daily presents a wealth of opportunities for extracting meaningful insights. This project explores the application of Latent Dirichlet Allocation (LDA), a powerful topic modeling technique, to uncover latent thematic trends in Twitter data or news articles. The methodology involves four main steps: data collection and cleaning, where tweets or news articles are retrieved and preprocessed; LDA modeling, where 10-15 latent topics are identified based on word co-occurrence patterns; visualization, where topic-term associations are represented through word clouds; and interactive exploration, allowing users to click on words and retrieve relevant tweets or articles, enhancing topic understanding. By leveraging LDA and Python's data processing capabilities, this project aims to provide a valuable tool for exploring thematic trends in social media or news content. The interactive nature fosters deeper engagement, enabling applications like sentiment analysis, trend monitoring, and content recommendation systems. Ultimately, the project demonstrates the utility of topic modeling in extracting meaningful insights from unstructured textual data.

II. Data Collection

A. Scrapping the Tweets from Twitter using NTScraper

The script begins by importing the required libraries: ntscraper for scraping tweets from Nitter instances, and wordcloud for generating word clouds (although it's not used in the provided code). It then defines a function called create_tweets_dataset that takes two arguments: username (the Twitter username to scrape tweets from) and no_of_tweets (the number of tweets to scrape). Inside the function, an instance of the Nitter class from the ntscraper library is created, and the scraper.get_tweets method is called with the provided username, mode="user" (to scrape tweets from a user's profile), and number=no_of_tweets (the number of tweets to scrape).

A dictionary data is created to store the tweet data, with keys for 'link', 'text', 'username', 'user', 'likes', 'quotes', 'retweets', and 'comments'. A loop iterates through the 'tweets' list returned by scraper.get_tweets, and for each tweet, the relevant data is appended to the corresponding lists in the data dictionary. The data dictionary is then converted into a Pandas DataFrame using pd.DataFrame(data).

Scraping has been done for multiple usernames such as Marques Brownlee, sama: - Sam Altman, AndrewYNg: - Andrew Coursera founder, Geoffrey Hinton: - Geoffrey Hinton, Yann LeCun:- Chief AI Scientist at Meta, karpathy: - Andrej Karpathy Director of AI @ Tesla, tim_cook: - Tim Cook CEO Apple, Mustafa Suleyman: - Mustafa Suleyman CEO, Microsoft AI, demishassabis: - Demis Hassabis Co-founder & CEO, ClementDelangue: - Co-founder & CEO,

B. Saving data to CSV

The DataFrame is written to a CSV file with the filename username_tweets_data.csv using the df.to_csv method. A message is printed to indicate that the scraping process is completed, and the CSV file is ready. Finally, the create_tweets_dataset function is called with the arguments "elonmusk" (the Twitter username) and 100 (the number of tweets to scrape). The purpose of this script is to scrape a specified number of tweets from a given Twitter user and store the tweet data (link, text, username, user, likes, quotes, retweets, and comments) in a CSV file for further analysis or processing

C. *Merging data frames into CSV*

- The script begins by importing the required libraries: pandas for data manipulation and analysis, and glob for finding pathnames matching a specified pattern. It then specifies the path to the directory containing the CSV files using the glob. glob function, which creates a list of csv_files containing the paths of all CSV files in the specified directory. An empty list dfs is created to hold the DataFrames for each CSV file.

- The script then iterates over the list of CSV files using a for loop. For each CSV file, the following steps are performed: the CSV file is read into a Pandas DataFrame using pd.read_csv(csv_file), and the DataFrame is appended to the dfs list using dfs.append(df). After reading all CSV files, the script merges the DataFrames in the dfs list along the rows (axis=0) using the pd.concat function, creating a single merged DataFrame merged_df. The ignore_index=True argument is used to ignore the existing row indices and create a new sequential index for the merged DataFrame.

- The path and filename for the merged CSV file are specified, and the merged DataFrame merged_df is written to a CSV file using the to_csv method. The index=False argument is used to prevent the row indices from being written to the CSV file. Finally, a message is printed to indicate that the merged CSV file has been saved. The purpose of this script is to combine multiple CSV files located in a specified directory into a single CSV file, which can be useful for consolidating data from different sources or processing large datasets that were split into multiple files.

- The merging process concluded with a confirmation message indicating the successful completion of the CSV file merger. This file serves as the primary data source for subsequent analyses and visualizations described in the report.

III. DATA CLEANING

Cleaning of the tweets

The script is designed to clean and preprocess tweets contained within a DataFrame. It defines a function, clean_tweet, which removes URLs, special characters, and numbers from each tweet, then tokenizes the text into individual words. It further processes these tokens by removing common English stopwords and lemmatizing the words to their base forms. The cleaned tokens are then joined back into a single string. This function is applied to the 'text' column of the DataFrame, resulting in a new column 'cleaned_text' that holds the preprocessed tweets. Any rows with empty or NaN 'cleaned_text' are subsequently removed to ensure only meaningful data is retained.

Following the cleaning process, the script tokenizes the cleaned text and uses Gensim's Dictionary class to create a dictionary of unique terms from the tokenized tweets. Finally, the cleaned and processed data frame is saved to a CSV file named cleaned_tweets.csv. This file can then be used for further analysis or model training. The script assumes that necessary libraries such as pandas, re, nltk, and gensim are imported and that required NLTK resources are downloaded.

IV. APPLING LDA FOR TOPIC MODELING & CREATING A WORD CLOUD BASED ON THE TOPIC STRENGTH

This script processes cleaned tweet data to perform topic modeling using Latent Dirichlet Allocation (LDA) and visualizes the topics with word clouds and bar plots. It begins by importing necessary libraries such as pandas for data manipulation, gensim for topic modeling, wordcloud for generating word clouds, plotly.express for visualization, and nltk for stopwords. The script tokenizes the cleaned tweets, creating a list of tokens for each tweet. It then creates a dictionary of unique terms from these tokens using Gensim's Dictionary class and converts the tokenized tweets into a bag-of-words representation, which serves as the input for the LDA model.

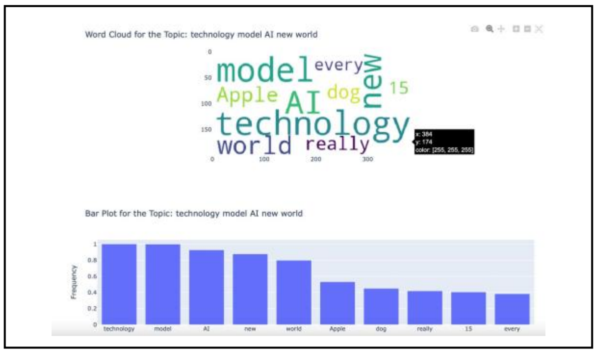
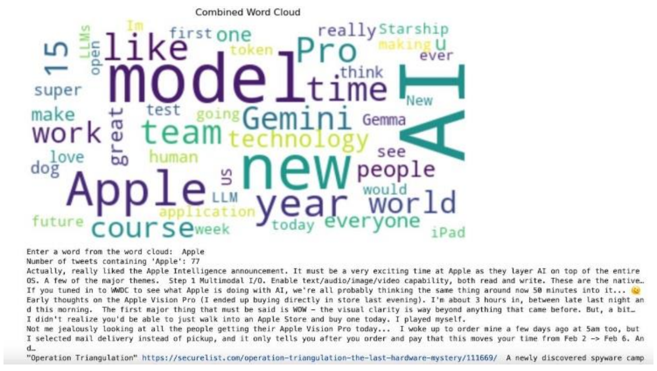


Fig. 1. Word cloud and bar graph

V. PRINTING RELATED TWEETS & NUMBER OF TWEETS FOR THE GIVEN WORD FROM THE WORD CLOUD



```
Enter a word from the word cloud:  Apple
Number of tweets containing 'Apple': 77
Actually, really liked the Apple Intelligence announcement. It must be a very exciting time at Apple as they layer AI on top of the entire
OS. A few of the major themes.  Step 1 Multimodal I/O. Enable text/audio/image/video capability, both read and write. These are the native…
If you tuned in to WWDC to see what Apple is doing with AI, we're all probably thinking the same thing around now 50 minutes into it... 😉
Early thoughts on the Apple Vision Pro (I ended up buying directly in store last evening). I'm about 3 hours in, between late last night an
d this morning.  The first major thing that must be said is WOW - the visual clarity is way beyond anything that came before. But, a bit…
I didn't realize you'd be able to just walk into an Apple Store and buy one today. I played myself.
Not me jealously looking at all the people getting their Apple Vision Pro today...  I woke up to order mine a few days ago at 5am too, but
I selected mail delivery instead of pickup, and it only tells you after you order and pay that this moves your time from Feb 2 -> Feb 6. An
d…
"Operation Triangulation" https://securelist.com/operation-triangulation-the-last-hardware-mystery/111669/  A newly discovered spyware camp
```

Fig. 2. Combined word cloud with description of tweets

Figure 2 shows a combined word cloud from the top words across all topics identified by an LDA model and enables users to display tweets related to specific words from the word cloud. It aggregates word frequencies from each topic into a single dictionary and creates a word cloud using the `WordCloud` library, excluding common stopwords. The word cloud is plotted using `matplotlib.pyplot`.

The script also defines a function to display tweets containing a specified word, prompting the user to input a word from the word cloud and then showing the number of relevant tweets and their content, allowing for interactive exploration of the tweet dataset based on the word cloud visualization.

Acknowledgment

I would like to extend my sincere gratitude to everyone who contributed to the successful completion of this project. Special thanks to my instructor for their invaluable guidance and support throughout the project. I am also grateful to my peers for their constructive feedback and collaboration.

Conclusion

In conclusion, this project successfully preprocesses and analyzes tweet data by leveraging LDA for topic modeling and visualizing the results through word clouds and bar plots. By cleaning the tweets, removing stopwords, and lemmatizing the text, the data is prepared for effective modeling. The LDA model uncovers underlying themes within the tweets, and the visualizations provide intuitive insights into the most significant words for each topic. Additionally, the interactive feature of displaying related tweets based on user-selected words from the word cloud enhances the exploration of the dataset, making the project a comprehensive tool for understanding and analyzing tweet content.

References

[1] Buchmueller, A., Kant, G., Weisser, C., & Saefken, B. (2021). Twitter Topic Modeling and visualization for R. https://cran.r-project.org/web/packages/Twitmo/Twitmo.pdf

[2] Ganeshmorye. (n.d.). GitHub - ganeshmorye/twitter_topic_modeling: Topic modeling on tweets from users in India over a 2-year period to discover trending topics and discussions. GitHub. https://github.com/ganeshmorye/twitter_topic_modeling.

[3] Negara, E. S., Triadi, D., & Andryani, R. (2019). Topic Modelling Twitter Data with Latent Dirichlet Allocation Method. 2019 International Conference on Electrical Engineering and Computer Science (ICECOS). https://doi.org/10.1109/icecos47637.2019.8984523