MLTweet-Explorer

Abstract

The application focuses on clustering data to identify topics such as politics, sports, etc. It categorizes tweets into meaningful groups.

It involves preprocessing tweets by cleaning (spell check and removing common words) and vectorizing. To categorize tweets, we used k-means clustering. To visualize data for better understanding, we used techniques such as 2D SVD and word cloud.

The K-Means clustering successfully divides tweets into clusters, visualized and represented by word clouds which highlight significant terms.

Identified topics offer insights into user accounts preferred tweets, targeted engagement and provide a basis of understanding online conversation.

Introduction

1. What is your application?

The application is a Twitter data clustering tool that uses natural language processing techniques such as text cleaning, TF-IDF vectorization, and KMeans clustering, to group tweets into different topics. Once these tweets are grouped, the program will find groups and topics similar to these clustered tweets and recommend them to a user. The goal/idea is to revamp twitter's recommendation system based on someone's specific interests by recommending key groups rather than specific people, allowing them to network and meet like minded people with fellow interests.

2. What are the assumptions/scope of your project?

While the tool is designed for Twitter data, this can be generalized to other social media platforms or text sources. The model assumes a consistent and relatively standard use of English language across tweets, and variations in language styles, slang, or domain-specific jargon may affect clustering accuracy. The user accounts and tags/groups that are recommended after clustering and interest identification are based on predefined mappings to identified topics and assume that these accounts are relevant and appropriate for engagement within the context of each topic.

3. Justify why is your application important?

Marketing teams can utilize the tool to gain insights into prevalent twitter topics, it can help them to create targeted content and engagement strategies.
Brands can monitor tweets related to their name and products to see if a product is failing or not. Researchers can leverage the tool to analyze trends, sentiments, and discussions in specific domains, aiding academic studies, market research, and trend forecasting. During crises or emergencies, the application can help monitor and categorize tweets. Political Analysis is one of the best examples of this application's usefulness. Normal everyday people can leverage this system in an era of increasing loneliness to find shared hobby and interest groups similar to how

Facebook does it, and interact and chat with people like that, instead of simply following a relevant page. Encourages increased user interaction both with the site and each other and ensures continued and increased usage of the application by encouraging and fostering communication and interaction/engagement.

4. Similar applications

There are similar applications in the domain of text clustering and categorizing topics on social media. But there are also uniqueness in this application such as, inclusion of spell checker in preprocessing pipeline make this work unique, it improves quality of text data by correcting spelling errors before clustering to improve accuracy. The application dynamically categorizes tweets into topics without relying on predefined labels, offering flexibility for analyzing diverse datasets with evolving themes. The tool places a specific emphasis on community engagement, making it a valuable resource for community managers, marketers, and individuals seeking to actively participate in online discussions.

5. Adjustments to part 1

- **Dataset Change:**
  The initial planned dataset consisted of tweets with metadata like the account username, timestamp, and the tweet's text content. However many of these datasets were missing critical information like tags, so a compromise was made where several different datasets were merged containing 2 simple columns, the text, and the ID classification, representative of the tweets main topic for cluster model training purposes.
- **Algorithm Change:**
  Initially, we considered using Multinomial Logistic Regression for our project, but after careful consideration, we opted for **unsupervised** machine learning with the **KMeans** algorithm. This decision was driven by the need to uncover natural patterns in tweet data without predefined categories. The shift from Multinomial Logistic Regression to KMeans allows our system to autonomously identify inherent groupings, providing flexibility and enhancing our ability to find meaningful insights within the tweets; flexible grouping instead of attempting to box vaguely fitting information into specific pre-programmed boxes that are far from all-encompassing.

Methodology

1. Design/pipeline

- **Data Ingestion:**
  Read Twitter data from an Excel file containing the 'Tweet_Text' column.
- **Text Preprocessing:**
  Clean the text by removing URLs, non-alphabetic characters, and converting to lowercase.
  Tokenize the text into words.
  Apply a spell checker to correct spelling errors.
  Remove stop words.
- **Vectorization:**
  Utilize TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to represent the cleaned text as numerical features.

- **KMeans Clustering**
  Apply KMeans clustering algorithm to the vectorized text.
  Determine the optimal number of clusters or use a predefined number (e.g., 5 clusters).
- **Dimensionality Reduction for Visualization:**
  Employ Truncated SVD (Singular Value Decomposition) to reduce the dimensionality of the TF-IDF matrix for 2D visualization.
- **Cluster Labeling:**
  Assign cluster labels to each tweet in the dataset.
- **Topic Description and User Account Suggestions:**
  Analyze important terms in each cluster and describe the main topic.
  Suggest user accounts based on the identified topics.
- **Visualizations:**
  Generate 2D scatter plots using the reduced dimensions for cluster visualization.
  Create word clouds for each cluster to visually represent important terms.
- **Output and Save Results:**
  Display the resulting DataFrame containing cleaned text, cluster labels, topic descriptions, and suggested user accounts.
  Save the DataFrame to an Excel file for future reference.

2. Dataset

Source of Raw Dataset: As mentioned previously the dataset we used was a combination of several other datasets merged together, all of which were found on Kaggle. The datasets are:

1. **Politics**: Tweets discussing political events, candidates, policies, or issues.
2. **Sports**: Tweets related to sports events, teams, athletes, or sports news.
3. **Technology**: Tweets covering technology trends, software, or innovations.
4. **Health**: Tweets focusing on health, medical topics, or healthcare news.
5. **Finance:** Tweets about financial markets, investments, stocks, or economic trends.
   https://www.kaggle.com/datasets/sulphatet/twitter-financial-news

Data Preprocess techniques:

- **Url removal:** Url doesn't contribute and can introduce noise.
- **Non alphabetical character removal:** special characters, symbols, and numeric digits may not add substantial meaning to the text for clustering purposes. Lowercasing ensures consistent representation
- **Tokenization:** Tokenization breaks down the text into individual words, facilitating further processing and analysis at the word level
- **Spell checking and stop word removal:** stop words (the, is , a, etc.) don't have significant meaning and can create noise
  Preprocessing is needed to enhance text quality of data, removing irrelevant data is important for improving the accuracy of the algorithm. Lowercasing and tokenization is necessary for consistent and standardized representation of text. Preprocessing reduces noise in data and ensures that clustering algorithm only focuses on relevant data.
  using raw data would create clusters of irrelevant words which are meaningless, necessitating the preprocessing of data

3. Model training

- Input is Term Frequency- Inverse Document Frequency Matrix
- **KMeans** clustering :
  input: TF-IDF matrix
  technique: Unsupervised machine learning
  Output: Cluster labels assigned to each tweet
- **SVD** (singular value decomposition):
  input: TF-IDF matrix
  technique: Dimensionality reduction technique
  Output: Reduced-dimensional representation of the TF-IDF matrix

4. Prediction

The prediction process involves replicating the same pre-processing steps and vectorization process that were applied during training. The new example is then fed into the trained KMeans clustering model to obtain a cluster label. If visualization is part of the application, the TF-IDF matrix is also transformed using the same Truncated SVD that was used during training.

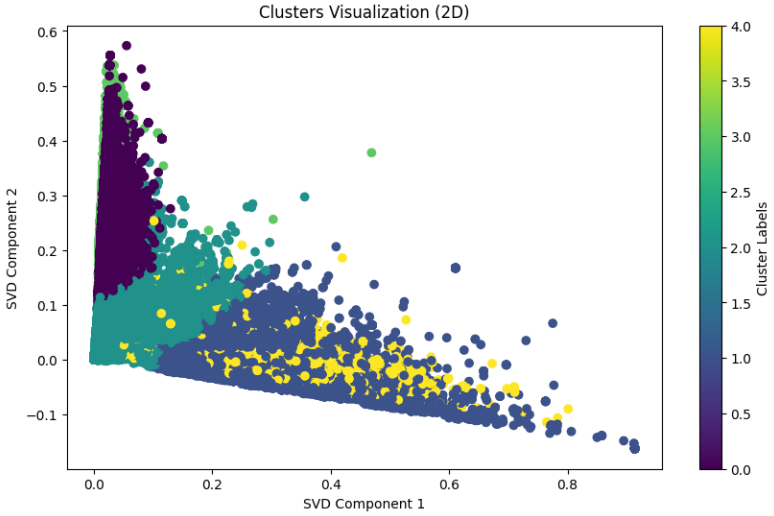
Results

1. Evaluation

Task for Evaluation:
Clustering Accuracy: Evaluate the accuracy of the clustering results by comparing the assigned cluster labels with true labels
Visualization Effectiveness: Assess the effectiveness of the 2D visualizations generated using Truncated SVD by analyzing how well they represent distinct clusters.

2. Results

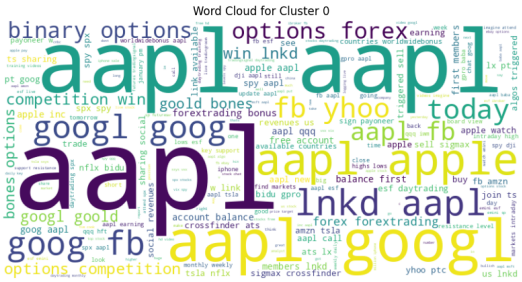
We opted for a clustering approach with five clusters. The visual representation indicates that three of these clusters exhibit clear distinctions, while the remaining two are more dispersed among the first three. This dispersion is attributed to the similarity in their content, leading to some inefficiencies in the clustering process.



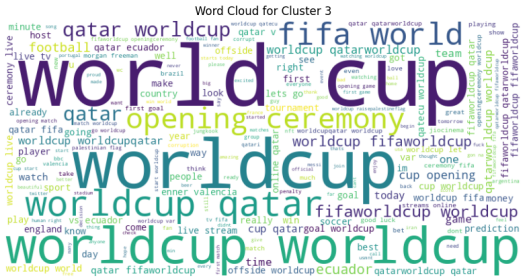
Following the execution of K-Means clustering with five clusters, we obtained the subsequent

clusters along with their key terms:

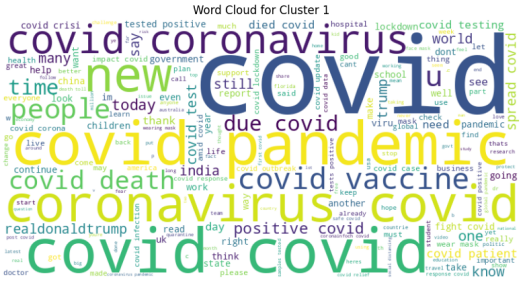
Cluster 1 - Important Terms: - aapl - apple - fb - goog - googl



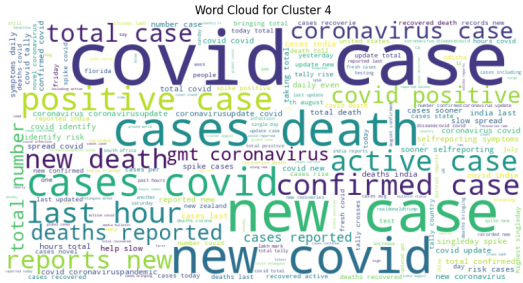
Cluster 4 - Important Terms: - worldcup - qatar - cup - fifaworldcup - world



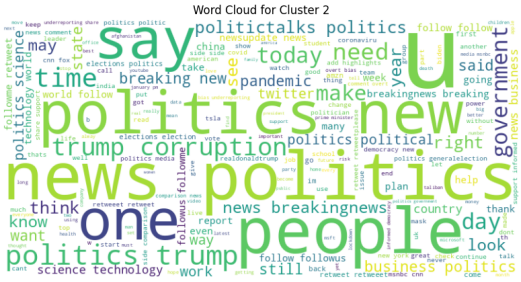
Cluster 2 - Important Terms: - covid - coronavirus pandemic - people - vaccine



Cluster 5 - Important Terms: - cases - coronavirus - new - deaths- people



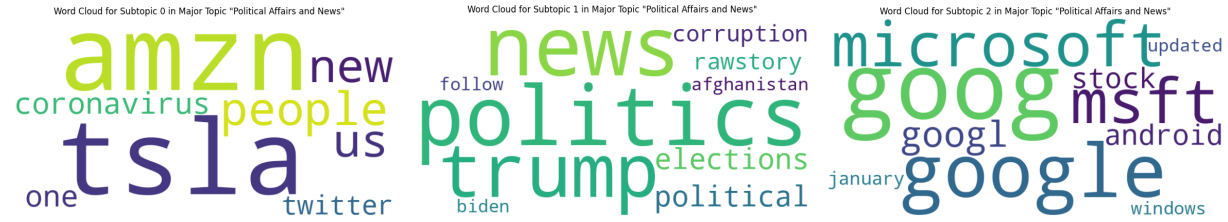
Cluster 3 - Important Terms:- politics - news - trump - political - elections



Following that, we assigned manual labels to them, categorizing them as Financial Markets and Stocks, Public Health and Pandemic Updates, Political Affairs and News, Global Football Events, Technology Trends, and Innovations.

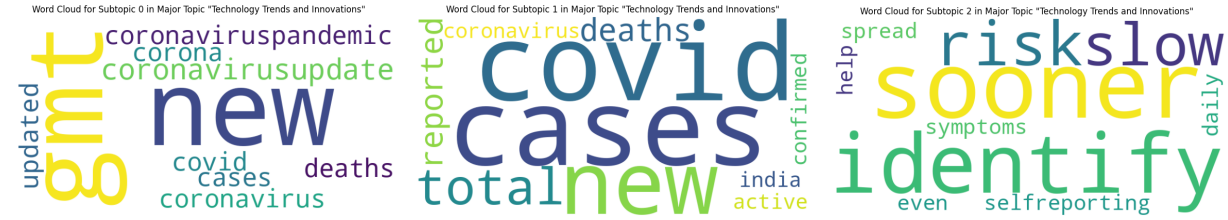
3. New Results After Peer Review

- We experimented with further dividing each cluster into smaller clusters to be able to improve the suggestion based on the context of the tweet. For example, instead of suggesting general Tech Company Stocks or Politics accounts to follow, we aimed to suggest further, to the point accounts to follow based on the individual interests within each cluster such as Google Stocks or Democrat Party accounts.
- Word Cloud for Subtopics in Political Affairs and News

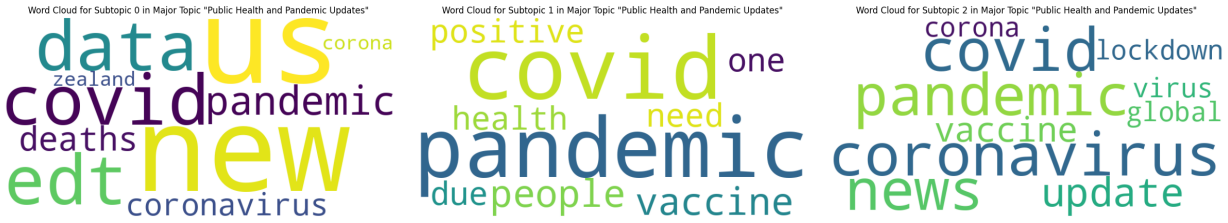


Word Cloud for Subtopics in Financial Markets and Stocks

Word Cloud for Subtopics in Technology Trends and Innovations



Word Cloud for Subtopics in Public Health and Pandemic Updates



Word Cloud for Subtopics in Global Football Events

- This experiment was not as conclusive as we expected due to the non-perfect division of clusters. The results showed us that we need to work on more precise clustering techniques. The problem could also stem from the dataset containing overlapping or related topics, and it's possible that the collected data is skewed towards certain subjects due to imbalanced dataset sizes when merging.

Discussions

1. Implications

- Results suggest that goal has been achieved. Successfully categorized tweets into different clusters. The inclusion of word remover in text pre-processing likely played a

crucial role in improving the quality of the text data and subsequently enhancing clustering accuracy.

2. Strengths

- Spell checking: The spell-checking feature ensures that the text data is cleaned and corrected, reducing the impact of typos and misspellings on subsequent analyses.
- Topic Categorization: The KMeans clustering algorithm excels at categorizing tweets into distinct topics or clusters based on content similarity. It's effective in identifying patterns and inherent groupings within the dataset.
- User Account Suggestions: The function that suggests user accounts based on identified topics provides a practical and relevant feature, enhancing user engagement and interaction within specific thematic clusters.
- Visualization: The code includes visualizations, such as 2D scatter plots of clusters using Truncated SVD, and word clouds for each cluster. These visualizations aid in understanding and interpreting the results of the clustering algorithm, making the output more accessible and informative.

3. Limitations

- Language: Application assumes English language text, improvement is needed to extend to multiple languages. variations in language styles, slang, or domain-specific jargon may affect clustering accuracy
- Optimization for Specific Domains: Depending on the dataset, the tool may benefit from domain-specific optimizations to enhance performance in niche topics or industries.
- Twitter API: Unfortunately the strict API changes recently imposed by twitter make it difficult to access too many of an account's tweets without having to pay, and tightened security means third party API's are a no-go. This inability to access someone's tweets in real time is a severe handicap to the goal of the project

4. Future directions

- Multilingual support: expand tool's support to handle text in multiple languages.
- User Interaction Features: Integrate more user interaction features, allowing users to customize and explore the data interactively, potentially enhancing the overall user experience.
- As aforementioned some bypass to the

6. Adjustments to part 3

- Attempted utilizing DBScan for clustering, but opted against proceeding with it due to processing time constraints.
- Identified additional stop words that nltk library failed to provide and removed them in the preprocessing.
- Changed the file types from xlsx to csv for faster read and write. Increased the speed from 20 seconds to 2 seconds
- We experimented with different values for the edit distance to find the right balance between correcting typos and preserving the original words. Spell.distance = 2 was thought to be a good starting point, as it allows for slightly more flexibility in correction.

However, our final observation of the result and the efficiency of the algorithm showed us that Spell.distance = 1 was more correct and efficient.

- Given that we are analyzing tweets that might include a lot of typos, it makes sense to set a relatively higher edit distance to capture and correct those typos effectively. Users often make casual and informal language use in tweets, leading to a higher likelihood of typographical errors.
- Further improvements made to data preprocessing to get cleaner results independent of regional slang and jargon

Additional Questions

1. What are the feedback that you found useful from the peer evaluation?

    - The majority of the feedback we got was mostly report based, primarily on how we could improve our evaluation section. The key takeaways from there were to add visuals, and be more specific in terms of numericals as to what categories we're labeling and testing our data under.
    - Another report based feedback we got was elaborating more on brought up terms, such as TF-IDF in the very beginning, and elaborating further on the preprocessing stage
    - Past that and heading into the feedback based on the actual project:
        - Multilingual support
        - Further improve and work on the outlined limitations
        - Factor in for region based slang and jargon

2. What changes did you make based on the feedback from peer evaluation?

    - We proceeded to create SVD Cluster Visualizations to illustrate the relationships between clusters. Additionally, we incorporated the previously generated word clouds for main topics into Deliverable 3. We introduced a new experiment, aiming to further divide each cluster into smaller clusters, and generated word clouds for them. This allowed us to observe the true nature of our cluster divisions and identify any errors.
    - In this exploration, we observed that some subclusters were more closely related to other main topics. For instance, within the Political Affairs and News topic, there were two subclusters related to Financial Markets and Stocks. Similarly, within the Technology Trends and Innovations topic, two subclusters seemed more connected to Public Health and Pandemic Updates. Overall, the subclusters demonstrated a purity of 65% within each main cluster.
    - The Subtopics, along with their visualizations, are presented in the Results section of the report.
    - Preprocessing and TF-IDF were clarified earlier so readers with no background would not be so confused, and as aforementioned significant improvements and clarifications were made to the Evaluation section
    - Text based limitations were improved on as outlined in "Adjustments to part 3" section of Discussions, encompassing aforementioned limitations and slang and jargon translation issues

References

Politics: Tweets discussing political events, candidates, policies, or issues.
https://www.kaggle.com/datasets/kaushiksuresh147/political-tweets

2. Sports: Tweets related to sports events, teams, athletes, or sports news.
https://www.kaggle.com/datasets/tirendazacademy/fifa-world-cup-2022-tweets

3. Technology: Tweets covering technology trends, gadgets, software, or innovations.
https://github.com/zfz/twitter_corpus/blob/master/full-corpus.csv

4. Health: Tweets focusing on health, wellness, medical topics, or healthcare news.
https://www.kaggle.com/datasets/gpreda/covid19-tweets

5. Finance: Tweets about financial markets, investments, stocks, or economic trends.
https://www.kaggle.com/datasets/sulphatet/twitter-financial-news