Trend Analysis and Content Generation on X and TikTok

AI Dreamers

February 2025

Project Report submitted for the Hackathon Competition
February 2025

Contents

List of Figures

Abstract

This project explores trend analysis and content generation on social media platforms, specifically X and TikTok. Utilizing data scraping, sentiment analysis, and machine learning algorithms, we aim to detect emerging trends and generate relevant content. Our solution provides insights for businesses, marketers, and content creators, enabling them to stay ahead in the dynamic social media landscape. Future work includes enhancing the AI agent and expanding platform coverage.

Acknowledgments

We would like to express our heartfelt gratitude to the organizers and the sponsoring enterprise of this hackathon for providing us with this invaluable opportunity. Their efforts in creating such a challenging and stimulating competition have been truly inspiring. We are also deeply thankful to the mentors who provided insightful information sessions and continuous mentorship throughout the project. Their guidance and support played a crucial role in the successful completion of our work.

Chapter 1

Introduction

1.1 Motivation

The rapid growth of social media platforms has led to an exponential increase in user-generated content. Understanding and leveraging trending topics is crucial for businesses, content creators, and marketers. This project aims to bridge the gap between raw data and actionable insights through AI-powered trend analysis.

1.2 Use Cases

The Viral Trend Analysis challenge is the first task in this hackathon. It involves developing AI-powered tools capable of scraping data from social media platforms like X and TikTok to detect trending topics in real time. Additionally, these tools will generate relevant videos, text, and images aligned with the latest trends, which can then be posted on various social media accounts to maximize engagement.

Chapter 2

Technical Tools

2.1 Programming Language

Python is a flexible and powerful programming language, widely used for trend analysis due to its scalability, real-time processing capabilities, and strong support for natural language processing (NLP). Its extensive ecosystem of libraries makes it an ideal choice for working with social media data.

2.1.1 Why Python for Trend Analysis?

Python's versatility, simplicity, and vast collection of libraries provide significant advantages in data scraping, processing, and analysis. Below are some key reasons why Python stands out:

- **Rich Ecosystem of Libraries:** Python offers a comprehensive set of tools tailored for data handling, analysis, and visualization:
  - **Pandas:** Efficient data manipulation and analysis.
  - **NumPy:** Numerical computations.
  - **Matplotlib & Seaborn:** Data visualization.
  - **Scikit-learn:** Machine learning models and data processing.
  - **NLTK:** Natural language processing (NLP) for analyzing social media text.
  - **Tweepy:** Accessing the Twitter (X) API.
  - **TensorFlow & PyTorch:** Deep learning and AI models when needed.
- **Scalability and Real-time Processing:** Python supports multi-threading and asynchronous programming, making it suitable for handling large-scale social media data in real time.
- **Strong Community Support:** A vast community of developers continuously enhances Python's capabilities, ensuring access to the latest advancements in AI and trend analysis.

Chapter 3

Pipeline and Architecture

3.1 X

3.1.1 Trend Analysis Workflow

The trend analysis pipeline follows a structured process to detect and leverage emerging trends on social media platforms. The workflow consists of the following key steps:

1. **Data Scraping:** Relevant data is collected from social media platforms such as X and TikTok.
2. **Data Storage:** The scraped data is stored in a structured format like csv or unstructered format lik json for further processing.
3. **Data Preprocessing:** The data is cleaned, normalized, and prepared for analysis.
4. **Analysis:** Key patterns and insights are extracted from the processed data.
5. **Dashboard Creation:** A visualization dashboard is generated to monitor trends effectively.
6. **Trend Detection:** Emerging topics and viral content are identified based on predefined criteria.
7. **Prediction:** Future trends and content opportunities are forecasted using AI-driven models.
8. **Post Generation:** AI generates relevant text, images, and videos based on detected trends.
9. **Posting to a plateform:** The generated content is automatically posted on X to maximize engagement.

Workflow Diagram

The following figure illustrates the complete trend analysis pipeline:

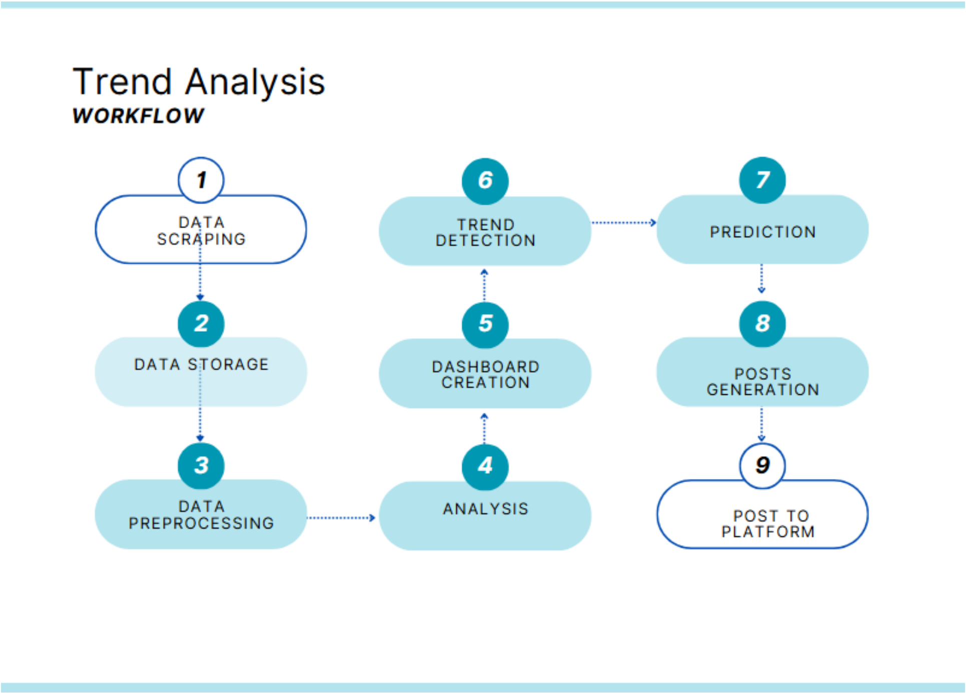

Figure 3.1: Python-based Trend Analysis Workflow

Architecture

3.1.2 Step 1&2: Tweet Extraction and Dataset Building

The first step in the trend analysis pipeline is extracting tweets and constructing a structured dataset. This process is essential for collecting real-time insights from X (formerly Twitter) and identifying emerging trends. The extraction is performed using the `Tweepy` library, which interacts with the Twitter API to fetch relevant tweets based on predefined keywords related to technology, artificial intelligence, cybersecurity, and other trending topics.

The system employs a query-based approach to retrieve tweets containing terms such as *AI*, *machine learning*, *blockchain*, and *cybersecurity*. To ensure high-quality data, the script applies various filtering conditions, such as removing retweets and limiting the results to English-language tweets. The extracted tweets include metadata such as the timestamp, username, tweet text, hashtags, retweet count, like count, follower count, and user description.

The dataset is incrementally built and stored in a CSV file named `trending_tweets.csv`, ensuring that each entry captures crucial engagement metrics. Additionally, a text-cleaning function removes unnecessary characters, mentions, and URLs to improve the quality of the extracted data. The system also integrates a directory creation mechanism for storing extracted tweets and related media files. In scenarios where API rate limits are encountered, the script efficiently handles pagination, enabling continuous data retrieval while adhering to Twitter's access restrictions.

Furthermore, an optional module allows for fetching global trending topics dynamically. The extracted data is processed and saved in an Excel file, where hashtags, external links, and media attachments are categorized for further analysis. This structured data serves as the foundation for downstream tasks such as trend detection, predictive modeling, and content generation.

3.1.3 Step 3 : Data Preprocessing

In this phase, we clean and preprocess the dataset by handling missing values, removing duplicates, and transforming textual and numerical data. We replace missing values in key columns such as `Followers_count`, `Tweet_text`, and `Description` with default values. The timestamps are converted into structured features like hours, days, and weekends. Additionally, hashtags are parsed, and sentiment analysis is performed using an emoji-based sentiment dictionary. These steps enhance data quality and prepare it for further analysis.

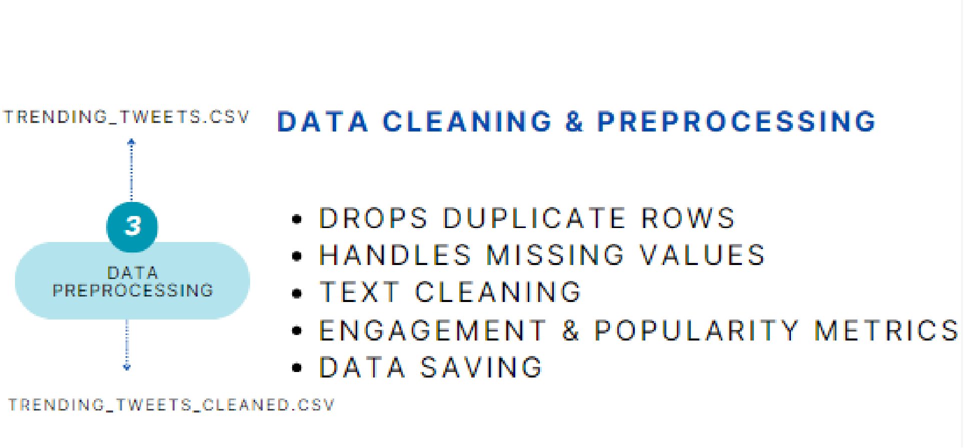

Figure 3.2: Data Preprocessing

3.1.4 Step 4 : analysis

This Step analyzes trending tweets using sentiment analysis, visualization, and forecasting techniques. The goal is to process tweet data, analyze engagement, detect influencers, and predict viral trends.

Data Processing and Sentiment Analysis

- **Pandas** (`pd.read_csv`) for loading and processing the dataset.
- **Hugging Face Transformers** (`RobertaForSequenceClassification`) for sentiment classification using the pre-trained `cardiffnlp/twitter-roberta-base-sentiment` model. .

Data Visualization

- **Matplotlib and Seaborn**: Used for visualizing sentiment distribution, scatter plots, and heatmaps.
- **WordCloud**: Generates word clouds for positive, neutral, and negative tweets.
- **Plotly**: Creates interactive charts to show engagement and influencer impact.

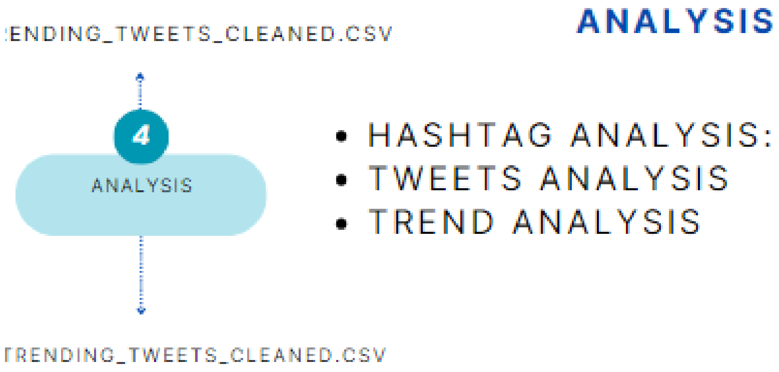

Figure 3.3: Data Preprocessing

Trend Analysis and Forecasting

- **Facebook Prophet**: Used for time-series forecasting to predict viral tweet trends.
- **Plotly Express**: Generates visual reports in HTML format.

3.1.5 Step 5 : Dashboarding

Twitter Trend Dashboard is the 5th step. It is an interactive application developed with Dash and Flask. It allows to display visualizations of Twitter trends in the form of interactive graphs and static images. The figures are generated and stored in a dedicated directory, then displayed dynamically via a web interface. The application also uses CSS styles to improve aesthetics and user experience.

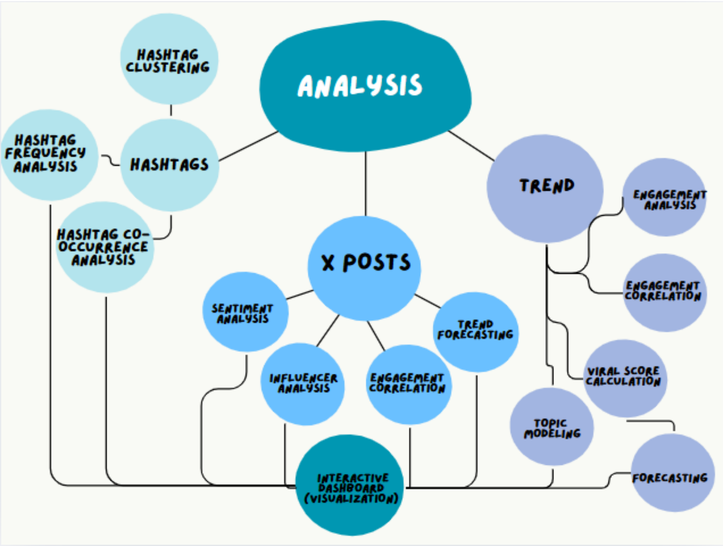

Figure 3.4: Dashboard

3.1.6 Step 6: Trend Detection

Trend detection involves identifying patterns and emerging topics within the dataset. This step utilizes clustering, keyword extraction, and time-series analysis to pinpoint viral discussions. The primary goal is to detect rapidly growing topics before they reach peak popularity.

Techniques Used

- **TF-IDF (Term Frequency - Inverse Document Frequency):** Used for keyword extraction from tweet text.
- **Latent Dirichlet Allocation (LDA):** A topic modeling approach to group tweets into thematic clusters.
- **K-Means Clustering:** Helps in grouping tweets with similar engagement and content.
- **Time-Series Analysis:** Evaluates tweet frequency over time to detect emerging trends.
- **Named Entity Recognition (NER):** Identifies important entities such as brands, influencers, or products trending within a dataset.

Justification for Techniques

- **TF-IDF:** Provides a numerical representation of word importance, improving keyword extraction accuracy.
- **LDA:** Helps uncover latent topics within large tweet collections.
- **K-Means:** Enables efficient clustering of similar tweets based on engagement, sentiment, and content.
- **Time-Series Analysis:** Offers insights into trend evolution, allowing proactive engagement strategies.
- **NER:** Enhances brand monitoring by identifying key players in trending topics.

3.1.7 Step 7: Prediction

Once trends are detected, predictive modeling is applied to forecast their future trajectory. This helps in determining the longevity and potential virality of emerging topics.

Techniques Used

- **ARIMA (AutoRegressive Integrated Moving Average):** A time-series forecasting model used for predicting tweet frequency trends.
- **Facebook Prophet:** A powerful forecasting tool for identifying seasonal and long-term trends.

- **Neural Networks (LSTMs):** Used for deep learning-based trend prediction, capturing complex patterns over time.
- **Regression Models:** Linear and logistic regression models help quantify the impact of different factors on trend evolution.

Justification for Techniques

- **ARIMA:** Well-suited for short-term forecasting of tweet volume and engagement.
- **Prophet:** Handles time-series data with missing values and seasonality effectively.
- **LSTMs:** Captures long-range dependencies in sequential tweet data for accurate trend prediction.
- **Regression Models:** Provide interpretable insights into how engagement metrics influence trend growth.

3.1.8 Final Step: Content Generation

The final step in our pipeline involves AI-driven content generation. We use transformer-based language models like GPT-2 to create engaging posts based on the detected trends.

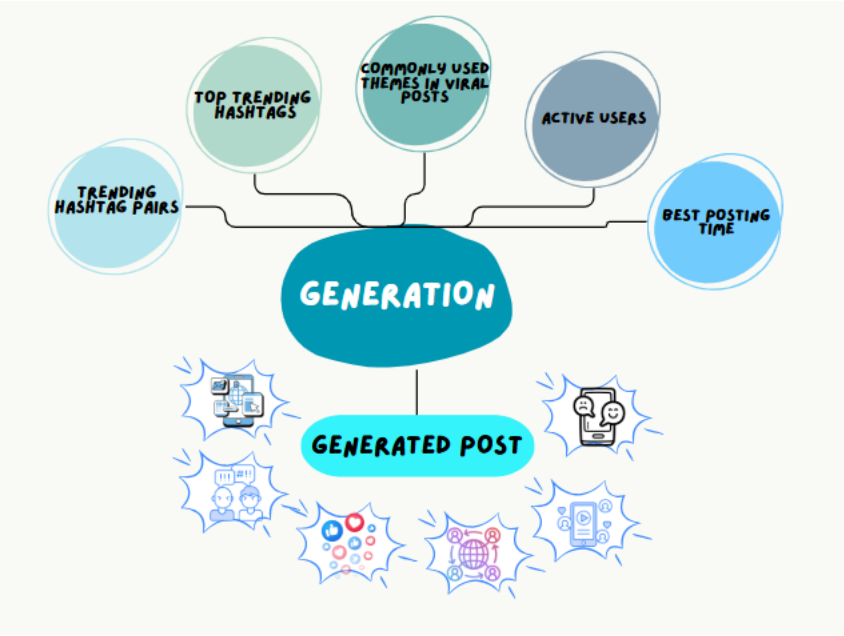

Figure 3.5: Generating Content

3.2 TikTok

3.2.1 Scraping Data from TikTok and Dataset Building

The analysis of viral trends on TikTok begins with the extraction of trending videos and the construction of a structured dataset. This process is carried out using an unofficial

TikTok API in combination with Playwright, a headless browser automation tool that simulates real user interactions to bypass platform restrictions. The system continuously retrieves trending content by mimicking user behavior on the “For You Page” (FYP), fetching batches of 30 videos per request.

Each extracted video is enriched with metadata, including author details (username, profile picture, verification status), engagement statistics, music information, and more. The data is stored in a MongoDB collection, where an empty comments field is initialized for each entry to be populated in a subsequent step. To maintain data freshness, a scheduling mechanism triggers the scraping script every 10 minutes.

The dataset is further enhanced by incorporating user comments, which provide valuable insights into audience engagement and sentiment. The system queries MongoDB for videos without comments and subsequently retrieves comments for each video. Extracted comments include text, author details, engagement metrics, language information, and more, which are then appended to their corresponding video records.

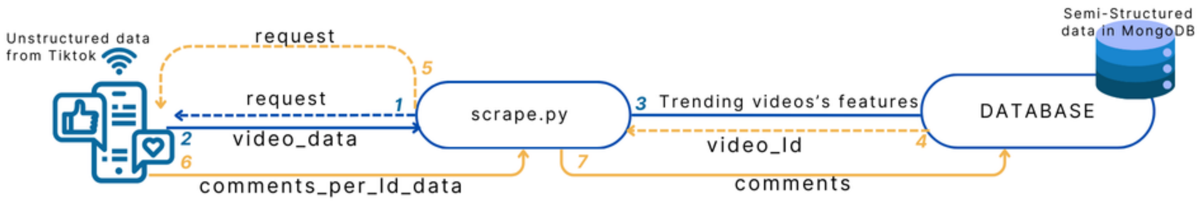

Figure 3.6: Data collection and storage Workflow

3.2.2 Data Processing and Feature Engineering

Data processing and feature engineering play a crucial role in ensuring the quality of the dataset for further analysis.

Data Cleaning: Data-cleaning techniques are employed to remove inconsistencies, special characters, handle duplicates, text normalization, filter irrelevant terms hat could introduce noise or bias in the analysis ...

Feature Engineering: Feature engineering involves selecting or creating the most informative features for analysis. Only the features relevant to each specific analysis are retained and more informative features are created.

3.2.3 Analysis

3.4.3.1 Author Popularity and Virality Detection Using Engagement Metrics

Understanding author influence on social media platforms is critical for trend analysis and content optimization. This analysis leverages key engagement metrics to quantify author popularity, assess virality, and identify top-performing influencers.

To measure author popularity, a popularity score is computed based on a scoring formula that assigns a weighted contribution, giving

- 60% importance to follower count
- 40% to heart count to balance audience reach and engagement levels.

- 10% boost for verified accounts.

To further evaluate influencer's impact, engagement rate and trend score are introduced:

- **Engagement Rate**: Measures audience interaction through metrics like likes, comments, and shares relative to the number of followers.
- **Trend Score**: Captures an author's virality by integrating multiple engagement factors. This score assigns weights to different metrics:
  - 50% weight to engagement rate (reflecting audience interaction),
  - 30% weight to video count (indicating content production volume),
  - 20% weight to share count (representing content reach and virality).

This approach ensures a balanced evaluation of both an author's overall popularity and their short-term virality potential and allow us to identify the top 10 most influential authors.

3.4.3.2 Hashtag Analysis for Trend Detection Using NLP

- **Hashtag Frequency Analysis** To identify emerging trends, hashtag frequency is analyzed, highlighting the most commonly used terms over a specified period. Term Frequency-Inverse Document Frequency (TF-IDF) is employed to quantify the relative importance of hashtags, ensuring that widely used but contextually significant terms are prioritized over generic ones.
- **Co-occurrence Analysis** Examining hashtag co-occurrences provides insights into content relationships and thematic groupings. Frequently appearing hashtag pairs reveal how different topics intersect, offering a strategic advantage in understanding audience interests and engagement patterns.
- **Clustering & Topic Modeling** To further analyze hashtag relationships, K-Means clustering is utilized to group similar hashtags based on contextual usage. Additionally, Latent Dirichlet Allocation (LDA) is applied to uncover underlying thematic structures within hashtag usage.

3.4.3.3 Video Duration and Posting Time Analysis

The goal of this analysis was to explore two key factors influencing the virality of TikTok videos.

- **Best Video Duration** Identifying the ideal length for a video is based on the relationship between video duration and engagement metrics like the number of views, shares, comments, and likes.
- **Best Time to Post** Examining how engagement varied depending on the time a video was posted.

3.4.3.4 Sentiment Analysis on Comments

- **Translation** Since the dataset contains multilingual comments, the first step was to translate the comments into English to ensure uniformity in sentiment analysis. The `deep-translator` library was employed to handle the translation process, with a fallback mechanism to retain the original text in case translation failed.

- **Sentiment analysis** The `cardiffnlp/twitter-roberta-base-sentiment` model was selected in this phase. This model, based on the RoBERTa architecture, is fine-tuned specifically for sentiment analysis in tweets. It has been shown to perform well in social media text due to its ability to capture the nuances of informal language, emojis, and slang commonly used in these platforms. Its robust performance on benchmark datasets for sentiment analysis makes it a suitable choice for analyzing TikTok comments, which often contain similar informal language and expressions.

  By leveraging this model, sentiment scores were computed for each comment, helping to categorize the comments based on their emotional tone, thus providing insights into user reactions (reply, like) to the content.

3.4.3.5 Audio Processing

To enhance engagement, trending Tiktok audios were analyzed and processed in order to generate new audio content later.

Trending TikTok videos were downloaded, and their audio tracks were extracted for further analysis. The extraction process ensured that only relevant, high-engagement audios were considered. The Whisper model was used for automatic speech recognition (ASR) to transcribe these audio tracks, enabling textual analysis. The use of Whisper for transcription provided high-accuracy text extraction from audio, enabling precise content analysis.

3.2.4 Data Visualization

Interactive visualizations are integrated for each analysis to facilitate the exploration of trend evolution and content alignment.

3.2.5 Trending Content Generation

- **Caption Generation for Optimized Engagement**

  In this section, the goal is to create engaging captions for trending content, which can significantly impact engagement rates on platforms like TikTok.

  To achieve this, the model `gemini-1.5-pro-latest` is employed for natural language generation (NLG). This model excels in producing coherent and contextually relevant captions by leveraging large-scale pre-trained data.

- **Video Generation**

For video generation, the `stabilityai/stable-diffusion-2-1` model is utilized to create dynamic and visually engaging videos. This model, based on the latest advancements in generative AI, is capable of producing high-quality visuals through diffusion techniques. By leveraging the model's ability to generate frames and seamlessly combine them into a coherent video, it enables the production of creative and visually appealing video content.

- **Content Generation using RAG**

  To generate relevant and engaging content, a Retrieval-Augmented Generation (RAG) model (`facebook/rag-sequence-nq`) was employed. This model combined information retrieval with text generation to produce contextually rich and relevant scripts. The transcribed audio served as input to the model, which then generated a refined textual output for the next stage. RAG was chosen due to its ability to retrieve relevant knowledge and generate coherent text, ensuring that the generated speech aligned with viral trends

  The generated text was then converted into speech using `gTTS` (Google Text-to-Speech). This step ensured that newly created audio content aligned with current trends while maintaining natural and engaging speech synthesis. gTTS was used for its efficiency in converting text into high-quality speech, allowing for rapid audio generation to be used in future content creation.

Chapter 4

Results and challenges

4.1 Results

The project demonstrates a complete end-to-end pipeline—from data collection and preprocessing through trend detection, forecasting, and AI-driven content generation. The results include a clean, feature-rich dataset, comprehensive visual analyses, forecast insights for optimal engagement timing, and automatically generated content ready for social media posting.

4.2 Challenges

• Restrictions from Twitter: Twitter only provides 1of tweets generated matching the query made by the extractor. Even though the number of tweets received is adequate, the guarantee of it being representative is still a problem to be addressed

• Data Acquisition and API Limitations: Social media platforms impose strict API rate limits and data access restrictions, making it challenging to collect large-scale, real-time data.

• Reliable Internet Connection: For continuous data extraction, the underlying Internet connection should be reliable. Even if access to the Internet is always available, the bandwidth provided has to be more or less constant for any accurate scientific calculations. If the bandwidth cannot be maintained, any normalization techniques have to be deployed.

Chapter 5

future work and potentials

5.1 Future Work

To further enhance our framework, several extensions can be implemented. Future improvements may include integrating additional social media platforms and incorporating user preferences to tailor trend analyses and content recommendations. Advanced search functionalities could allow users to focus on specific topics of interest, while refinements in forecasting models would improve the accuracy of trend predictions. Enhancing the content generation module to produce more personalized and context-aware outputs is also a promising direction. Together, these enhancements will pave the way for a comprehensive, intelligent system for continuous, real-time trend analysis and content generation. as well we will work on enhancing existing features such as Web portal that shows the dashboard and use better AI modeles that needs preminum access In addition , by applying for apis request that are much better we might have a better chance at getting richer datasets Finally, automation and cloud deployment will be essential for ensuring scalability, reliability, and seamless real-time processing of trends and content generation.

Chapter 6

conclusion

Social media platforms such as X and TikTok offer rich, real-time data for trend analysis and content generation. In this project, we successfully implemented an end-to-end pipeline that extracts live streaming data, preprocesses it, and performs comprehensive analyses including sentiment evaluation, topic modeling, and trend detection. Our framework generates interactive dashboards that display trending topics, key engagement metrics, and forecasted viral trends. Furthermore, transformer-based models have been employed to generate engaging content that aligns with the latest trends. The system provides valuable insights for businesses, marketers, and content creators, enabling them to optimize their digital strategies and stay ahead in a dynamic social media landscape.