Analyzing Trending Movie Trailers

Ömer Faruk Zeybek
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University
Istanbul, Turkey
150220743

Ulaş Polat
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University
Istanbul, Turkey
150200322

***Abstract*—The rise of digital media has transformed the way audiences discover and engage with movie trailers. This project leverages big data analytics to identify trending movie trailers by analyzing audience interactions such as comments and sentiment. The system integrates data collection, natural language processing, and visualization to offer real-time insights into audience preferences, addressing gaps in traditional platforms like IMDb and Rotten Tomatoes. The proposed solution aims to improve user decision-making and engagement with movie content.**

***Index Terms*—Movie trailers, big data, natural language processing, sentiment analysis, Spark NLP.**

CONTENTS

I. INTRODUCTION

The rise of digital media has transformed how people discover and consume movie trailers. With countless trailers released daily, users often struggle to decide which ones to watch. Popular trailers from the past with millions of views also resurface periodically, making it difficult to distinguish trending content. Traditional platforms like IMDb and Rotten Tomatoes rely heavily on editorial selection, which can introduce biases and fail to reflect real-time user engagement. For example, IMDb, owned by Amazon, may unfairly prioritize trailers for Amazon Prime Video's exclusive content on its trailer pages, potentially overshadowing other trending trailers and creating an unbalanced representation. Furthermore, these platforms lack insights derived from user comments, which could provide valuable context and opinions.

This project aims to address these gaps by leveraging big data analytics to offer real-time insights into trending movie trailers. By analyzing thousands of YouTube comments daily, the system identifies popular trailers and provides comment-based insights to help users decide whether to watch or skip a trailer. The solution highlights trending trailers on a dedicated website, offering an intuitive platform for users to explore the most engaging content. Through this approach, our goal is to improve the decision-making process and to improve the overall user experience.

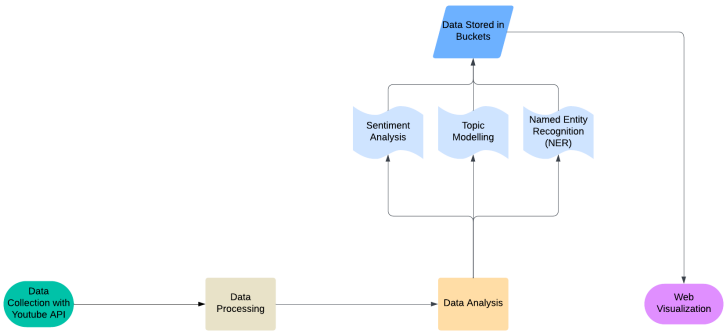

Fig. 1. Project Workflow Diagram

II. DATA COLLECTION AND PROCESSING

The data collection and processing pipeline was designed to systematically collect, clean and prepare movie trailer metadata and user comments for analysis. The pipeline is composed of two main scripts: 'datacollection.py' to collect trailer metadata and 'getcomments.py' for fetch comments from YouTube. The data collected are initially stored in JSON

format. To optimize storage and facilitate big data processing, a separate script is used to convert these JSON files into Parquet format. The resulting Parquet files are then uploaded to Google Cloud Storage for further processing on Google Cloud Dataproc. This semi-automated approach was adopted to streamline the testing phase, with plans to fully automate the pipeline on cloud services in future iterations.

A. Data Sources

B. Data Sources

The primary data source for this project is YouTube. Using the YouTube Data API, trailers are searched based on predefined keywords, such as "official trailer" and "teaser trailer," while ensuring the inclusion of trailers only from target movie studios like Warner Bros., Marvel Entertainment, and Netflix. This filtering guarantees that the data is relevant and avoids duplication. Metadata such as video titles, view counts, like counts, and publication dates is collected, prioritizing the most viewed trailer for each movie.

Additionally, up to 200,000 comments for each trailer are fetched using the YouTube Data API. The process is designed to handle new and existing comments separately, ensuring that only unique and latest comments are collected. This approach helps maintain an up-to-date dataset for analysis.

C. Storage

The processed data, including trailer metadata and user comments, is first stored locally in JSON format. The data is then converted to Parquet format due to its efficiency in storage, compression, and compatibility with Spark DataFrames, which are essential for big data processing. The Parquet files are subsequently uploaded to Google Cloud Storage buckets, providing reliable, scalable, and accessible storage. Since Spark jobs are executed on Google Cloud Dataproc, the data stored in Google Cloud Storage ensures seamless integration and efficient processing.

Currently, this semi-automated workflow facilitates testing and validation of the pipeline. In the future, the process will be fully automated by deploying the necessary components on cloud services, enabling the continuous collection, conversion, and storage of data without manual intervention.

III. Analysis Techniques

The analysis pipeline employs advanced Natural Language Processing (NLP) techniques to extract meaningful insights from user comments on YouTube movie trailers. Using state-of-the-art models and frameworks, the system processes large-scale text data to perform tasks such as sentiment analysis, named entity recognition (NER), and topic modeling. These techniques provide a comprehensive understanding of audience opinions, trending topics, and the overall reception of trailers.

A. Natural Language Processing (NLP)

NLP is central to this project, enabling the extraction of actionable insights from unstructured text data. The system utilizes Spark NLP, a powerful library for large-scale NLP tasks that is fully integrated with Apache Spark for distributed processing. Pretrained pipelines from Spark NLP are employed for tasks such as:

- **Sentiment Analysis**: Identifies whether a comment expresses positive, neutral, or negative sentiment, providing a sentiment distribution for each trailer.
- **Named Entity Recognition (NER)**: Extracts key entities (e.g., people, organizations, or objects) mentioned in the comments to highlight trends and popular topics.
- **Topic Modeling**: Uses Spark NLP for preprocessing and Spark MLlib's implementation of Latent Dirichlet Allocation (LDA) to group comments into thematic topics, offering deeper insights into audience discussions.

1) SparkNLP and Spark: The project leverages Spark NLP for its scalability, speed, and ability to handle massive datasets in a distributed environment. Key details include:

- **Pretrained Pipelines**:
  - The 'recognize_entities_dl' pipeline is used for NER.
  - The 'sentimentdl_use_twitter' model with Universal Sentence Encoder is used for sentiment analysis. These models ensure high accuracy and performance without extensive fine-tuning.
- **Topic Modeling with Spark MLlib**:
  - The project employs Spark MLlib's Latent Dirichlet Allocation (LDA) for topic modeling.
  - The parameters are optimized to ensure a meaningful clustering of topics based on comment data.

By combining Spark NLP's cutting-edge models with Spark's distributed architecture, the system efficiently processes millions of comments, enabling real-time analysis of audience feedback and emerging trends.

2) Sentiment Analysis: The sentiment analysis process uses the Spark NLP library and its pre-trained models to classify user comments on movie trailers as positive, negative, or neutral. Using Spark's distributed processing capabilities, the system efficiently handles large-scale datasets stored in Parquet format. The analysis pipeline is summarized as follows:

1) **Pipeline Components**:
   - **Document Assembler**: Converts raw text into document format for processing.
   - **Universal Sentence Encoder (USE)**: Generates sentence embeddings, enabling semantic analysis of text.
   - **SentimentDL Model**: We selected the pretrained sentimentdl_use_twitter model because YouTube trailer comments are a form of social media comments, and this model is specifically designed to handle sentiment analysis for social media text.
2) **Process Workflow**:

- Comments are loaded from the latest Parquet files stored in Google Cloud Storage (GCS).
- The Spark NLP pipeline processes comments in batches to ensure scalability and efficiency.
- Each comment is classified with a sentiment label (positive, neutral, or negative).

3) **Sentiment Statistics**:
   - The system calculates sentiment distribution (e.g., percentages of positive, neutral, and negative sentiments) for each trailer.
   - Examples of each sentiment category are extracted to provide illustrative results.
4) **Scalability and Results Storage**:
   - The results, including sentiment distributions and statistics, are stored in JSON format and uploaded to GCS.
   - The distributed nature of Spark enables efficient processing of millions of comments in a reasonable time frame.

This approach ensures accurate, scalable, and seamlessly integrated sentiment analysis within the big data pipeline.

3) Named Entity Recognition (NER): The Named Entity Recognition (NER) process uses Spark NLP's pretrained pipeline, 'recognize_entities_dl', to extract and classify named entities from user comments. The workflow for NER is described below:

1) **Pipeline and Model**:
   - The 'recognize_entities_dl' pipeline is loaded to extract entities and their corresponding types (e.g., PER for persons, ORG for organizations).
   - The pipeline processes batches of comments, ensuring the scalability and efficient handling of large datasets.
2) **Process Workflow**:
   - Comments are read from Parquet files stored in GCS.
   - Comments are processed in chunks, where each chunk is passed through the NER pipeline.
   - The pipeline returns tokens annotated with entity types, and entities are extracted and aggregated across all comments.
3) **Entity Analysis**:
   - The extracted entities are counted and categorized according to their types.
   - A function identifies the top entities by frequency and groups them by type for a deeper analysis.
4) **Results and Insights**:
   - The results include the total entities identified, the top entities, and their respective counts and types.
   - Entities are grouped by type (e.g., persons, locations) to provide a clear breakdown of the data.
5) **Scalability and Results Storage**:
   - Entity extraction is performed in a distributed manner using Spark's RDD and DataFrame APIs.
   - Final results, including entity summaries and statistics, are saved as JSON files in GCS for easy access and further analysis.

This ensures accurate and scalable entity recognition, enabling meaningful insights into audience discussions.

4) Topic Modeling: The topic modeling process leverages the Latent Dirichlet Allocation (LDA) implemented in Spark MLlib to uncover thematic structures in user comments on movie trailers. The workflow for topic modeling is detailed below:

1) **Text Preprocessing**:
   - Spark NLP processes text data with:
     - **Document Assembler**: Converts raw text into structured document format.
     - **Tokenizer**: Breaks text into individual tokens.
     - **Normalizer**: Cleans tokens by lowercasing and removing punctuation.
     - **Stop Words Cleaner**: Retains meaningful tokens by filtering out common stop words.
     - **Finisher**: Produces an array of cleaned tokens.
2) **Vectorization**:
   - The processed tokens are converted into numerical vectors using Spark's 'CountVectorizer'.
3) **Latent Dirichlet Allocation (LDA)**:
   - LDA is applied to identify latent topics. Parameters include:
     - **Number of Topics**: Set to 5 to effectively summarize the discussions.
     - **Vocabulary Size**: Limited to 500 words for relevance.
     - **Iterations**: Configured for convergence in 10 iterations.
     - **Learning Parameters**: Adjusted for efficient large-scale dataset handling.
4) **Results and Insights**:
   - Each topic is represented by the top 10 associated words.
   - Model performance is evaluated using metrics such as logarithmic likelihood and perplexity.
5) **Scalability and Storage**:
   - The process leverages Spark's distributed architecture.
   - The results, including topics and model metrics, are saved in JSON format in GCS.

This approach provides robust and scalable insights into the discourse of the audience.

B. Cluster Configuration

The cluster configuration for processing movie trailer comments and metadata is designed to handle large-scale data efficiently while remaining cost-effective during the testing phase. The current setup consists of:

- **Cluster Setup:**

  - **Master Node:** Manages the overall processing tasks and coordinates distributed jobs across the cluster.
  - **Worker Node:** Handles the bulk of the computational load, processing data in parallel.
- **Processing Time:**
  - The cluster configuration is capable of processing the entire pipeline, including sentiment analysis, entity recognition, and topic modeling, within approximately 30 minutes for the 250 megabytes of text.
- **Potential Improvements:**
  - **Scaling Nodes:** Increasing the number of worker nodes would allow for parallel processing of more data, reducing overall runtime.
  - **Using Better VMs:** Upgrading to more powerful virtual machines with higher CPU and memory resources would enhance performance and reduce bottlenecks.

This configuration ensures an effective balance between performance and resource usage during development and testing, with clear avenues for optimization as the project scales.

IV. Website

In this section, we discuss the development of a web interface that will present the ranked trailer list in a dynamic and interactive format. This web interface, built with Flask, will allow users to view the most recent trailer rankings based on audience interactions such as views, likes, comments, and sentiment analysis. The trailer list will be updated daily, ensuring that the information displayed is always current and relevant.

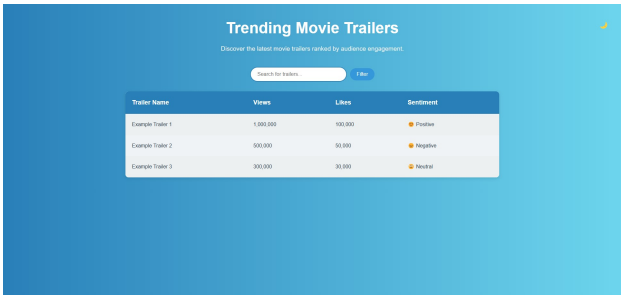

Fig. 2. Alpha Version of the Website (see Appendix)

- **Flask Web Interface:** The website will be built using Flask, a lightweight and flexible web framework in Python. Flask allows us to efficiently serve the dynamic trailer ranking table to users, with real-time updates. It will provide the functionality to display the ranked trailers, show detailed information about each trailer, and provide users with a search or filter option to explore trailers based on specific criteria, such as release date, popularity, or sentiment.
- **Dynamic Table:** The trailer list will be presented in a dynamic table, where each row represents a trailer along with its rank, title, sentiment breakdown and engagement metrics (e.g., number of views, likes, comments). This table will automatically refresh every day, ensuring that the ranked displayed reflect the most recent feedback and interactions from the audience.
- **User Interaction and Filtering:** The interface will allow users to interact with the trailer list in various ways. They can filter trailers according to genres, sentiment, or engagement levels, allowing users to explore different aspects of the data. In addition, users will be able to sort the table by various metrics such as rank, sentiment score, or number of comments, offering an interactive and personalized experience.

This website will serve as a user-friendly platform to present the results of the trailer analysis, offering visitors a clear view of which trailers are trending and how they are performing based on audience feedback.

V. Results

A. Sentiment Analysis

Below is the sentiment distribution for the trailer `Karate Kid`:

- **Analyzed Comments:** 11,988
- **Sentiment Distribution:**
  - Positive: 7,800 (65.1%)
  - Neutral: 804 (6.7%)
  - Negative: 3,384 (28.2%)

This distribution indicates that most viewers expressed positive sentiments towards the trailer, with a smaller portion showing neutral or negative feedback.

Sentiment	Count	Percentage
Positive	7800	65.1%
Neutral	804	6.7%
Negative	3384	28.2%

B. Entity Recognition

For the trailer `28 Years Later`, the top entities recognized were:

- **Total Comments Processed:** 29,731
- **Top Entities:**
  - Cillian (Person) - 1,072 mentions
  - Murphy (Person) - 947 mentions
  - iPhone (Miscellaneous) - 457 mentions
  - Boyle (Person) - 324 mentions
  - Danny (Person) - 313 mentions

These results suggest that the trailer's audience is particularly interested in prominent figures such as actor Cillian Murphy and director Danny Boyle. The mention of "iPhone" may indicate product placement or technological references within the film.

Entity	Type	Count
Cillian	Person	1,072
Murphy	Person	947
iPhone	Miscellaneous	457
Boyle	Person	324
Danny	Person	313

C. Topic Modeling

For the trailer `American Primeval`, the following topics were detected:

- **Total Comments:** 424
- **Top Topics:**
  - Topic 0: see, looks, can't, Indians, Taylor
  - Topic 1: movie, one, I'm, good, wait
  - Topic 2: netflix, white, hate, man, bad
  - Topic 3: like, Netflix, looks, Hollywood, don't
  - Topic 4: people, show, land, Revenant, actually

The topics highlight various discussions, ranging from excitement about the movie's plot and characters to opinions about the platform Netflix. These themes suggest a mix of positive engagement and critical reactions.

Topic	Keywords
Topic 0	see, looks, can't, Indians, Taylor
Topic 1	movie, one, I'm, good, wait
Topic 2	netflix, white, hate, man, bad
Topic 3	like, Netflix, looks, Hollywood, don't
Topic 4	people, show, land, Revenant, actually

VI. Future Work

As we continue to develop this project, we have several exciting areas to explore and expand upon, aiming to enhance the system's capabilities and broaden its application to a variety of contexts.

- **Automating Workflows with Apache Airflow:** One of the major improvements we plan to implement is the automation of data pipelines and workflows using Apache Airflow. By integrating Apache Airflow, we can schedule and monitor complex workflows, ensuring that data collection, processing, and analysis tasks are performed efficiently and consistently without manual intervention.
- **Predicting Trending Trailer Fragments with MLlib:** We plan to use machine learning models to predict which parts of trailers (or promotional videos) are most likely to trend. By analyzing audience interactions such as views, comments, and likes, we will employ machine learning techniques, specifically MLlib, to identify high-impact trailer fragments. This could help content creators focus on the most engaging parts of their material to maximize audience engagement.
- **Including Upcoming Theater Trailers:** We aim to include trailers for upcoming theater releases, enabling the system to predict audience reactions and trends before these trailers are publicly available. This proactive approach will allow creators to gauge the interest of potential audiences in advance.
- **Expanding to Series, Sports Events, and Other Content:** In addition to movies, our system can be expanded to analyze trailers and promotional materials for TV series, sports events, or even live-streamed broadcasts. By adapting the analysis to different types of content, we can provide insights for a wider range of media industries. For example, by analyzing the engagement with promotional materials for sports events or live-streamed games, we can predict which moments or teams might generate the most fan excitement. This flexibility will allow the system to cater to various sectors, offering insights into audience behaviors for a wide range of events and media.

These enhancements will not only increase the scope of our project, but will also make it adaptable to diverse media forms, allowing us to generate dynamic, real-time insights into audience engagement and predict which content will capture attention across different domains.

VII. Conclusion

Integration of big data analytics with sophisticated natural language processing techniques has been shown to improve the way users find and interact with movie trailers. By extracting insights from large-scale datasets of audience interactions, such as comments and sentiments, the system gives users real-time insight into better decision making about which trailer to watch.

This has allowed scaling and efficiency in the processing pipeline, allowing meaningful insights into audience preferences by using Spark NLP for sentiment analysis, named entity recognition, and topic modeling. Furthermore, it develops a dynamic and interactive web interface that ensures that these insights are presented to the user in a very friendly way, enabling users to easily explore trending trailers.

Besides the obvious uses, some exciting prospects for further development of this project are threefold: automating data workflows using Apache Airflow, predicting high-impact trailer fragments with machine learning, and scalability to go beyond movie trailers into TV series and sports events. These features will make the system even more usable and rewarding for both users and content developers.

References

- Spark NLP: https://nlp.johnsnowlabs.com/
- Apache Spark: https://spark.apache.org/
- Google Cloud Platform: Dataproc, Cloud Storage https://cloud.google.com/
- Flask: https://flask.palletsprojects.com/
- YouTube Data API: https://developers.google.com/youtube/registering_an_application

APPENDIX

Fig. 3. Full View Alpha Version of the Website