TikTok Keyword System

A social media intelligence system

Name:	Venkata Sai Ganesh Chandu Bheesetty
Github:	GitHub Repository
Tableau Dashboard:	Tableau Dashboard

1 Summary

The **TikTok Keyword System** is a prototype end-to-end pipeline designed to extract, analyze, and visualize trending keywords from TikTok content at scale. It enables businesses, marketers, and analysts to monitor emerging topics by leveraging multi-modal keyword extraction from text, audio, and visual signals, followed by growth detection algorithms to identify the fastest-growing trends. This document provides a detailed breakdown of the architecture, methodology, metrics, and current prototype implementation.

2 Introduction

Social media platforms such as TikTok are dynamic ecosystems where trends emerge rapidly and influence digital marketing, production promotion, and cultural conversations. Tracking these trends in real-time provides businesses with a competitive edge, allowing them to adapt strategies, launch targeted campaigns, and measure audience engagement effectively.

The goal of this system is to build an automated pipeline that:
1. Collects TikTok videos at scale by web-scraping.
2. Performs multi-modal keyword extraction not limited to captions but extended to audio and video contents.
3. Identifies the fastest-growing keywords using statistical growth detection techniques.
4. Generate reports and dashboards with configurable filters such as category, geography, engagement thresholds and many more.
5. Supports both batches and scheduled execution using Airflow and containerized deployment.
The prototype has been developed with modular microservices for scraping, multimodal processing, trend detection, metrics evaluation, and visualization. All components are Dockerized and integrated into a CI/CD pipeline with GitHub Actions for automated testing and container registry deployment.

3 System Architecture Overview

At a high level, the system consists of the following layers:

1. Scraping layer:
   - Collects trending TikTok video metadata (captions, hashtags...) using TikTokApi.
   - Outputs JSON file (tiktok_trending.json)
2. Multi-Modal Processing Layer
   - Processes scraped data through piplines:
     - Text processing (captions/comments)

  - Audio transcription (planned: Whisper)
  - Visual keyword extraction (planned: OCR, image captioning)
- Produces structure keywords (keywords.csv)

3. Preprocessing
   - Takes keywords.csv file created from multimodal pipeline to generate some extra parameters for business analytics (keywords_clean.csv)
4. Database layer
   - TimescaleDB schema for long-term storage of keywords, growth statistics, and pipeline metrics.
   - Supports hypertables and continuous aggregation views for efficient queries.
5. Growth detection layer
   - Applied statistical analysis on keywords frequency over time
   - Identifies the fastest-growing terms across time windows, regions, categories, and engagement.
6. Orchestration layer
   - Apache Airflow DAGs automate the full pipeline execution (scraping → multimodal extraction → preprocessing → storage → growth detection → metrics collection).
   - Spark integration enables distributed keywords processing at scale.
7. Analytics & Reporting layer
   - Streamlit API for interative exploration which uses keywords_clean.csv file.
   - Tableau dashboards for business-focused reporting (category trends, engagement analysis).

4 Component & Technical Approach

4.1 Scraper

The purpose of this component is to fetch trending tiktok video metadata. This is built using Python requests with **unofficial TikTok endpoints**. This extracts id, username, url, caption, region, music, engagement metrics (likes, comments, shares, views). Saves results into tiktok_trending.json. This uses *ms_token* env variable while extracting data.

4.2 Multimodal Processing

This is the core innovation of the pipeline: extracting keywords from multiple modalities. In this project, we have extracted keywords from text (captions, comments), audio, video processing, image captioning.

Text Processing
a. Tokenization using **NLTK**.
b. Stopwords removal and filtering.
c. Returns top 20 keywords based on frequency.

Audio Processing
a. Speech-to-text with **OpenAI Whisper (small model)** for efficiency.
b. Language-auto-detection for multilingual captions.
c. Post-processing returning frequent keywords.

Music Processing
a. Extracting background music of each video.
b. Keywords extracted from this are authorName and title of music with music as a modality.

Visual Processing
Processes two types: OCR and image captioning.
a. Sources TikTok video frames and overlays.
b. **OCR** with **Tesseract** for on-screen text (titles, embedded hastags).
c. Image captioning with **nlpconnect/vit-gpt2-image-captioning** by using FFMPEG to extract frames.

Data Consolidation
Keywords from text, audio, visual merged per video entry.

4.3 Database (Timescale DB)

The database layer is implemented using TimescaleDB, which allows for efficient storage and time-series analysis of extracted keywords. The central table is ***keywords***, a hypertable that stores every detected keyword along with other parameters. This table serves as the ground truth log of the pipeline. On top of it, a continuous aggregate view ***keyword_hourly_counts*** groups keywords into hourly buckets by region and category, providing the basis for trend analysis.

Growth detection algorithms operate on these aggregates and persist their outputs into the ***trending_keywords*** table which stores the fastest-growing keywords along with their mentions, growth thresholds, and time windows. To monitor the health and efficiency of the pipeline, a separate ***pipeline_metrics*** table is maintained for keywords extracted, unique words extracted.

The workflow is straightforward: extracted keywords are first inserted into the ***keywords*** table, aggregates are periodically refreshed, growth detection jobs query these aggregates and save results to ***trending_keywords***, and metrics are logged into ***pipeline_metrics***.

4.4 Growth detection

The growth detection module is responsible for identifying keywords that are rapidly increasing in popularity over defined time windows. It operates on top of the continuous aggregates generated in TimescaleDB, such as the ***keyword_hourly_counts*** view. By comparing keyword frequencies across consecutive time buckets, the system computes relative growth thresholds and applies configurable thresholds to filter out only those terms with meaningful acceleration. Detected trends are then persisted in the ***trending_keywords table***, annotated with region, category, growth threshold, and the relevant time horizon (hourly, daily, or weekly). This process ensures that growth detection is both scalable and transparent, allowing downstream analytics or business teams to easily query trending topics across multiple dimensions.

4.5 Orchestration

The orchestration of the pipeline is handled via **Apache Airflow**, with DAGs (Directed Acyclic Graphs) coordinating the sequential and parallel execution of tasks. Typical DAGs include scraping TikTok videos, pre-processing multimodal content, inserting extracted keywords into the database, refreshing materialized views, and triggering growth detection jobs. This modular orchestration framework ensures fault tolerance and retry handling while enabling flexible scheduling (e.g., daily scraping runs combined with daily growth detection). For scalability and distributed workloads, **Spark** can be integrated as an execution engine for the heavier multimodal extraction tasks, although the pipeline also runs in a simplified mode without Spark for smaller-scale testing. Airflow thus provides the backbone of reliability and traceability across the entire system.

4.6 Dashboards

The visualization layer is powered by both lightweight **Streamlit** dashboards and exploratory **Tableau** dashboards. Streamlit provides a direct interface for monitoring pipeline outputs, enabling interactive filtering of keywords by region, category, or time horizon while pulling directly from the processed database tables. Tableau complements this by offering richer, business-focused analytics built on the cleaned dataset (keywords_clean.csv). For example, dashboards showcase category-wise trends, virality vs discussion, and engagement distributions, making the results accessible not just for developers but also for analysts and decision-makers. Together, these dashboards bridge the technical outputs of the pipeline with actionable business insights.

As part of this prototype, a sample dashboard was created using the processed ***keywords_clean.csv*** dataset. This dashboard demonstrates how extracted keywords can be transformed into business insights.

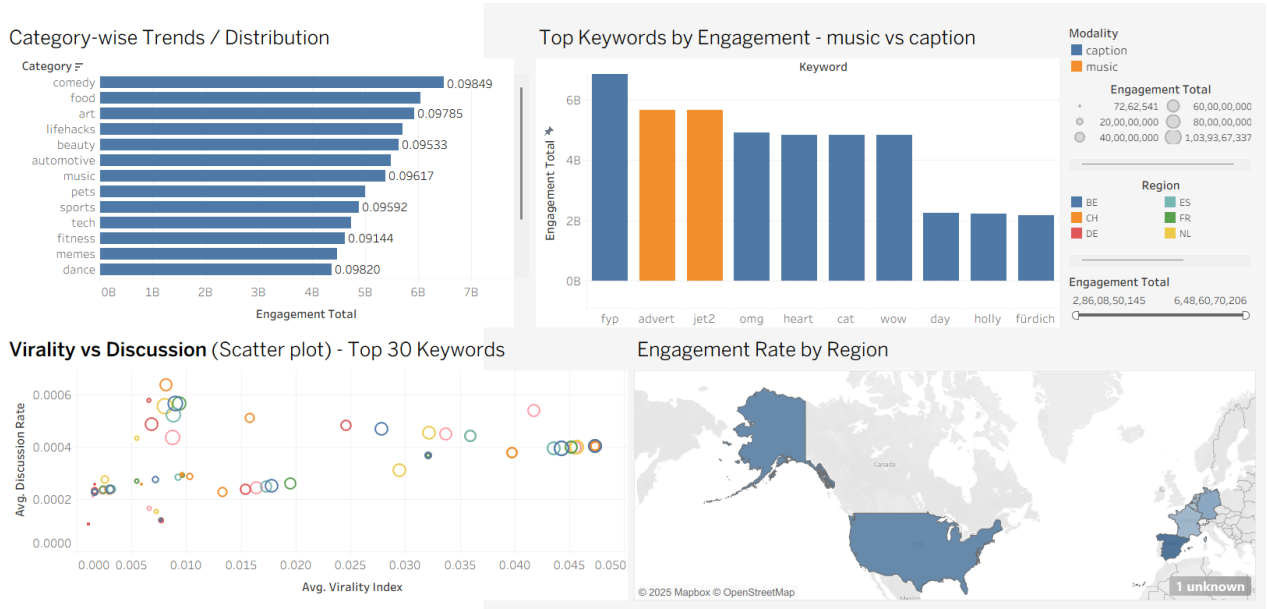

Figure 1: Sample Tableau Dashboard

5 AI coding Assitant Integration

During the development of the pipeline, AI coding assistants were integrated into the workflow to accelerate prototyping and reduce boilerplate implementation time. These assistants supported tasks such as generating Spark job templates, scaffolding Airflow DAGs, writing Dockerfiles for different components, and creating linting checks for critical functions. They were also instrumental in rapidly iterating database schema definitions and CI/CD configurations, ensuring alignment with best practices while minimizing manual debugging overhead. This integration not only improved productivity but also reduced the system development time.

6 Limitations and Future Work

While the system demonstrates a functional end-to-end pipeline, several limitations emerged during development that provide directions for future work.

Scraping Constraints:
The current scraper is capable of collecting captions and metadata, but access restrictions prevented large-scale automated extraction of video and audio streams. Although the codebase contains modules

for video and audio preprocessing, these could not be fully validated due to limitations in TikTok's data access policies. Future work could integrate official APIs or licensed datasets to enable full multimodal processing.

Scalability and Orchestration:
While Airflow DAGs and Spark jobs were integrated, the current system was tested on smaller workloads for feasibility. Scaling to the target of 15,000+ videos per day requires cluster-level resource tuning, workload scheduling, and distributed data validation mechanisms. Future iterations can leverage Kubernetes-based orchestration for improved elasticity.

Business Reporting and Analytics:
The Tableau dashboard built on ***keywords_clean.csv*** demonstrates proof of concept for keyword-based analytics. However, integration of live connections to the TimescaleDB instance would enable real-time dashboards with automatic refresh. Future dashboards could also incorporate growth detection alerts and anomaly detection visualizations.

Backtesting and Metrics:
A framework for recording pipeline metrics exists, but systematic backtesting across historical datasets has not yet been completed. Extending this would validate the reliability of growth detection algorithms under varying conditions (e.g., region, category, or engagement thresholds).

7 Conclusion

The **TikTok Keyword Trend System** successfully demonstrates a functional pipeline for scraping, multimodal keyword extraction, growth detection, orchestration, storage, and reporting. While constrained by access limitations for large-scale video and audio processing, the system showcases a modular architecture that can be extended to full multimodal analysis.

By combining automated data pipelines, distributed processing via Spark, time-series storage with TimescaleDB, and visualization with Tableau and Streamlit, this project lays the foundation for a scalable trend detection platform.

Future improvements will focus on unlocking video/audio modalities, optimizing the growth detection pipeline for scale, and expanding reporting capabilities to include real-time alerts and advanced analytics. With these enhancements, the system can evolve into a robust production-ready solution for identifying and analyzing TikTok keyword trends at scale.