

Sentiment Sphere: AI-Driven Sentiment and Topic Modelling Framework

Prof. Nitin Alzende, Om Chorghe, Kshitija Jadhav, Aditya Takale

Department of Computer Science & Engineering,

MIT ADT University, Loni Kalbhor 412201, India

{ omchorghe542004@gmail.com , kshitijaa05@gmail.com , adityatakale333@gmail.com }

Abstract:

This paper presents a Multi-Platform Sentiment Analysis Dashboard - a web-based system, which integrates the sentiment analysis of four sources of data, text input by users, CSV/TXT files uploaded by users, YouTube comment threads, and BlueSky social media. This app relies on an NLP pipeline that sums the results of both VADER and TextBlob in order to classify any given input as a positive, negative, or neutral sentiment. Additionally, it has a specialised Finance Pulse feature, to correlate real-time stock price data obtained via the yfinance library with a finance news sentiment score, as provided by the Fin BERT model. The outcome of sentiment analysis is visualised as an interactive React.js UI component of donut charts, stacked bar charts, 7-day candlestick graphs, and rows can be exported as tables. The empirical tests were performed with the help of a 3,151 entries Amazon Alexa reviews dataset and a 100-sentence-long custom dataset. Application backend is based on Python based Fast API.

Keywords: Sentiment analysis, natural language processing, VADER, TextBlob, Fin BERT, Fast API, React.js, YouTube API, BlueSky AT Protocol, data visualization.

1. INTRODUCTION

Sentiment Analysis is the process of deriving opinions, attitudes, and sentiments out of written text and is, perhaps, one of the most powerful uses of Natural Language Processing. In the current lives where text generated by users is so rampant on different social media sites, e-commerce websites, financial information websites and video sharing websites, solutions to integrate and analyze sentiment expressed in various platforms simultaneously are required.

Existing solutions are typically specialised towards their purpose: YouTube comment scraper, Twitter sentiment

analyser, or stock news aggregator. This division requires the analysts to alternate between various programs, overlooking the broader view, which may be very important in the analysis point. It is this need that our dashboard fulfils.

These five sources of data include manually typed text, importing CSV/TXT files, scraping YouTube comments with the YouTube Data API v3, post queries with AT Protocol using the Bluesky API, and financial headlines using the Finnhub API, with stock prices using yfinance. All the analysis will be displayed through a single React.js dashboard, which will run on a Fast API Python server.

2. RELATED WORK

Pang and Lee [1] were the pioneers in setting the standards of a seemingly simple rule-based system that would be as useful as more complex models in the analysis of lexicon in reviews. VADER, the creation of Hutto and Gilbert [2], is an effective lexicon-based framework of rules created to analyse text in social media, and can handle punctuation stress, emoji mood, and booster words effortlessly. Finally, Loria [3] incorporated to this list the TextBlob pattern scorer and polary.

Fin BERT is the BERT variant suggested by Liu et al. [4] to work on financial data, such as earnings calls, analyse opinions, and financial news, indicating the superiority of domain-specific transformers in comparison with their non-domain counterparts in the field of finance. The relationships between sentiment in the form of Twitter feed and shifts in the Dow Jones Industrial Average were established by Bollen et al. [5].

Earlier initiatives were made at integrating multiple platforms into dashboards, which has included Twitter and news data [6], but, has made no attempt that integrates Bluesky AT Protocol, YouTube comments, financial sentiment and general text analysis into a single web application.

3. OBJECTIVES

- Conduct a composite dashboard to conduct sentiment analysis on five disparate datasets.
- Use VADER + TextBlob ensemble on regular text and Fin BERT on financial text to implement sentiment classification.
- Connect live feeds with the Bluesky AT Protocol, the YouTube Data API v3 and Finnhub API.
- The challenge is to analyse the stock price variation in the context of the sentiment of the financial news post in Finance Pulse.
- Use dynamic charts like donut charts, bar graphs, candlestick charts, and confidence distribution to show the results of your analysis.
- Make sure that all processed datasets can be exported to CSV.
- The learners who are not tech-savvy and do not require knowing how to code to use the platform should find it easy to use.

4. SYSTEM ARCHITECTURE

The System utilises a client-server architecture with 3 tiers, mainly, React.js frontend, a Fast API backend written in Python, and external data APIs. The project structure in Visual Studio Code (with the Uvicorn server running) is shown in Fig. 1 below.



Fig. 1. The structure of the project and the Uvicorn backend that is running (localhost:8000)

A. Frontend (Client Layer)

The frontend is a single-page application (SPA) that was created with React.js and designed with Tailwind CSS. It has five route-based pages, namely: Home, Text Analysis, Bluesky Analysis, YouTube Analysis, and Finance Pulse. Recharts is used to generate charts; Axios is used to make the HTTP request.

B. Backend (Server Layer)

The backend is a Fast API, which is served with the help of Uvicorn (ASGI). The second layer is a backend which includes a router in each of the modules, which include the Bluesky analysis, finance analysis, text analysis, YouTube analysis and health modules. The second layer comprises sentiment_service.py/ word-frequency-service.py/bluesky-service.py/youtube-service.py/fiance-service.py. Environment variables are set out in .env.

C. External APIs

The YouTube Data API v3 allows you to retrieve comments and metadata of YouTube video videos, the Bluesky AT Protocol allows you to find posts with a specific word, and the Finnhub API provides you with time-stamped financial news. You can get stock OHLC data from yfinance.

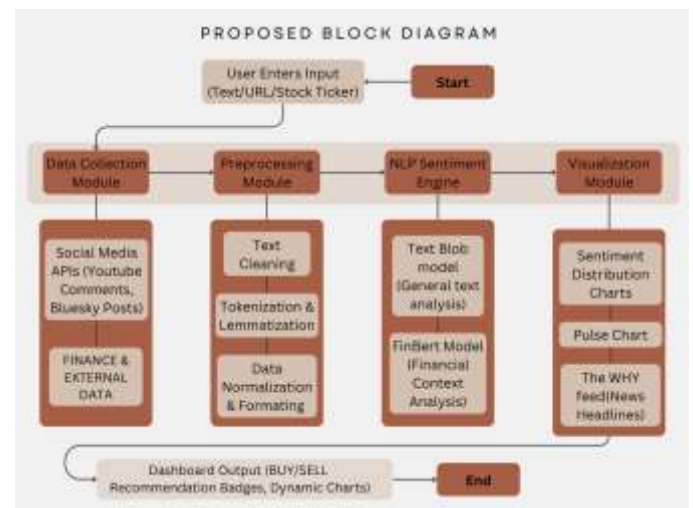


Fig. 2. The Proposed Block Diagram

The block diagram overview of the proposed system presented here is a preliminary sketch and is subject to change. The following block diagram shows the end-to-end data processing pipeline for the Sentiment Analysis Dashboard. The process is as follows:

- **User Input:** This process begins with an input from the user, which may be raw text, a YouTube link or a stock ticker.
- **Data Collection Module:** Depending on the input, the system will use integrated Social Media APIs like YouTube comments or BlueSky posts, or either Financial and External Data sources such as yfinance, Finnhub or uploaded CSVs.
- **Preprocessing Module:** The raw data is cleaned and standardized by text cleaning, tokenization, lemmatization: The AI models can process the

data accurately. Format data for normalization: The data is represented in a format suitable for processing by the AI models.

- **NLP Sentiment Engine:** The input text is redirected to the context-appropriate model, with a general text model (TextBlob ensemble), and a specialized financial text model (FinBERT model), which is used only for financial contexts.
- **Sentiment Module:** The output of the sentiment scores is structured and displayed in the visualization module, where interactive visualization elements such as Sentiment Distribution Charts, Finance Pulse Chart, and a "WHY" news feed are created.
- **Dashboard Output:** Lastly, for any user, the processed insights are displayed dynamically on the dashboard in the form of charts and the suggesting Buy/Sell tags based on insights.

5. TECHNOLOGIES USED

The backend of the system is developed with Fast API (Python), the frontend is developed with React.js and Tailwind CSS, and uses NLP models such as VADER, TextBlob, and Fin BERT to analyse sentiment. It also leverages APIs such as YouTube Data API and Finnhub to get data.

Table 1: Technologies Used in the System

Component	Technology
Backend framework	Fast API (Python)
ASGI server	Uvicorn
General NLP	VADER + TextBlob
Financial NLP	Fin BERT (Hugging Face)
Data processing	Pandas
Stock prices	yfinance
Financial news	Finnhub API
Frontend framework	React.js + Tailwind CSS
Charting	Recharts
HTTP client	Axios
Social API 1	YouTube Data API v3
Social API 2	Bluesky AT Protocol

6. METHODOLOGY

A. Data Collection

There are four significant ingestion routes of data. Text Analysis will operate on copied text or imported .txt/.csv files. YouTube Analysis gives an analysis of the video comments (up to 100 comment) by entering their URL of the YouTube Data API version 3. The Bluesky Analysis will implement the AT Protocol API to fetch the corresponding public posts by authenticating credentials. Finance Pulse compiles 7-day OHLC prices and stock news of a given stock ticker.

B. Preprocessing Pipeline

Unprocessed text is processed through a pipeline that is standardised: (1) processing to lowercase, (2) removal of special characters and URLs, (3) removal of stop words, (4) treatment of empty or null values and (5) string coercion of mixed CSV columns. This script word_frequency_service.py script generates a mapping of the tokens and provides the most N words.

C. VADER + TextBlob Ensemble

VADER model (Valence Aware Dictionary and Sentiment Reasoner) [2] gives a composite value at the range $[-1, +1]$. Any value equal to or greater than 0.05 is considered Positive with any value that is less than or equal to -0.05 is Negative. The values that are not in the extremes are said to be Neutral. The TextBlob library [3] provides another score of polarity, also in the same range.

D. Finance Pulse — Fin BERT

The Finance Pulse module replaces the generic ensemble model with a pre-trained Fin BERT [4] - a BERT model previously trained on financial data. The special language employed in the field of finance; Fin BERT removes this problem in traditional NLP models. Sentiment scores are assigned to each news article and the daily sentiment average is correlated with the corresponding Yahoo Finance stock price and plotted on a two-axis plot.

7. VISUALISATION TECHNIQUES

It uses six types of charts, which are implemented through Recharts:

- Donut chart — distribution of sentiments (Positive / Negative / Neutral counts)
- Bar chart — Top Words (counts of the top 10 non-stop tokens) frequency of the top-10 non-stop tokens

- Stacked bar chart — sentiment distribution by sentiment class (Low / Medium / High)
- Dual-axis line chart — closing price vs. average daily sentiment score overlay
- 7-day candlestick chart — OHLC data for the queried stock ticker
- Area chart — smooth 7-day closing-price trend

8. API INTEGRATION

A. YouTube Data API v3

The backend gets the video ID in the provided URL and makes a request to `commentThreads.list` to obtain comments (in batches of at most 100) and videos. Get the video details (title, author, views, likes) with a `List`. All the comments are processed by the VADER + TextBlob framework.

B. Bluesky AT Protocol

The `app.bsky.feed`. The `Posts API` annual search is called on the keyword entered by the user, and a fixable post limit. Authentication into the backend is done on the basis of app password. Text in the post is also extracted into the record format of the AT Protocol and scored using the sentiment analysis pipeline.

C. Finnhub + yfinance

The `yfinance` API loads 7-day OHLC values, whereas the `Finnhub Company News API` loads headlines on the same 7 days on a timelike basis. The headlines are then rated with `Fin BERT` model. Date: Closing prices and headline sentiment/day are compared by date and retracted into a single JSON object.

9. RESULTS AND SCREENSHOTS

A. Home Page

Fig. 3 shows the landing page presenting four module cards for analysis of different modules. The dark navy theme with gradient typography provides an accessible interface for non-technical users.



Fig.3. Sentiment Analysis Dashboard landing page

B.1 Text Analysis —Pasted Text

Fig. 4 demonstrates text analysis on a 100-sentence road-quality corpus. The donut chart shows 41 Positive, 38 Negative, and 21 Neutral entries.



Fig. 4. Text Analysis: donut chart for 100-sentence corpus

B.2 Text Analysis —Detailed Results

Fig. 5 Top Words bar chart and detailed results table per row with VADER compound score and TextBlob polarity for each sentence and ensemble label.



Fig. 5. Text Analysis: Top Words chart and VADER / TextBlob score table

C.1 File Analysis —CSV Upload

Fig. 6 File analysis for the Amazon Alexa review CSV file (3,151 rows): 2,614 Positive, 349 Negative, 188 neutral – reflecting actual consumer reviews for a widely-used product.



Fig. 6. File Analysis: Amazon Alexa CSV (3,151 entries)

C.2 File Analysis —Row-Level Results

Fig. 7 Row-level results for the Alexa dataset. Top word counts go up to 3,200, while product-related tokens such as "echo", "love", "great" confirm correct lexical analysis.



Fig. 7. File Analysis: detailed row-level sentiment and score table

D. Bluesky Analysis

Fig. 8 Forty-nine Bluesky posts analysed for "AI Software-based companies": 27 Positive, 17 Negative, 5 Neutral – suggesting a positive Bluesky narrative towards AI companies.



Fig. 8. Bluesky Analysis: keyword search sentiment overview

E.1 YouTube Analysis — Overview

Fig. 9 The YouTube module analyses Samay Raina's "STILL ALIVE" TV special (205,081 total comments). Sample results show 55 Neutral, 37 Positive, 8 Negative



Fig. 9. YouTube Analysis: sentiment distribution and video stats

E.2 YouTube Analysis —Confidence & Summary

Fig. 10 shows the stacked confidence distribution chart, a Video Summary card (public opinion: "mostly positive"), and individual comment sentiment labels.



Fig. 10. YouTube: confidence distribution and video summary card

F.1 Finance Pulse —Overview

Fig. 11 depicts the Finance Pulse component for GOOG (Alphabet Inc.). The two-axis finance pulse chart plots the average daily news sentiment against the closing price of the past seven days, and an AI determines a BUY signal.



Fig. 11. Finance Pulse: GOOG dual-axis sentiment/price chart and candlestick

F.2 Finance Pulse — Price Trend & News Feed

Fig.12 depicts the area price chart and “Why feed” card carousel for the past seven days, where news headlines are accompanied by Fin BERT sentiment score badges and confidence values.



Fig. 12. Finance Pulse: 7-day area trend and news sentiment card feed

The tech stack for analysis of Finance Pulse uses yfinance (Python library) for analysing Price Data and the Finhub API to analyse News Data. The NLP model uses the tool Fin BERT (via HuggingFace), and the Charts displaying analysis use Plotly.

10. DISCUSSION

The findings indicate that the use of a combination of VADER + TextBlob, with rules, produces sentiment classifications for various corpora, whether formal (product reviews) or informal (social media messages). This is evidenced by the Amazon Alexa data set (83% Positive) and the road quality data set (41% Positive, 38% Negative).

The incorporation of Fin BERT into the Finance Pulse module mitigates one of the main limitations of generic NLP solutions when applied in finance. The two-line graph of GOOG between April 23 and May 1, 2026 reveals that there is a direct relationship between a high sentiment and increase in price.

Bluesky sentiment analysis cannot search using chronological order due to the unsuitability of chronological order in the AT Protocol. The issues with the YouTube sentiment analysis include the following issues: The size of a batch merely allows them to gather 10,000 units in a batch in one day.

11. LIMITATIONS

- The context-awareness of a rule-based approach to NLP (VADER + TextBlob) is often misunderstood, as well as sarcasm, irony and code-mixed text are often misunderstood.
- This comes with certain disadvantages in YouTube Data API v3, Bluesky AT Protocol and rate limits on large-volume data acquisition at Finnhub.
- Computational methods The computational methods of Fin BERT model inferences are computationally expensive on a CPU, in order to scale inferences, inferences have to be accelerated on a graphics card using CUDA.
- The model does not necessarily deal with non-English or multi-lingual text.
- The AI BUY/HOLD/SELL recommendation is an algorithmic decision-making product, rather than an authorised product of financial advice.

12. FUTURE SCOPE

- Fine-tune BERT/RoBERTa models to obtain true sentiment analysis.
- To perform multi-platform analysis, use Twitter API v2 to perform sentiment analysis on Twitter/X.
- Use WebSocket to stream data to make a live sentiment analysis dashboard.
- To perform sentiment analysis on Hindi, Marathi, and even other Indian languages, take advantage of the capability of XLM-RoBERTa to support more than a single language.
- Host on AWS Lambda/ API Gateway (Backend) and Vercel (Frontend) such that anyone has access to it.
- Model: Add an Aspect-Based Sentiment Analysis (ABSA) feature, which can be used to analyze sentiment towards particular entities.
- Use PRAW to log into Reddit to provide more social sentiment data and in vast quantity.

13. CONCLUSION

The most suitable solution would be the Multi-Platform Sentiment Analysis Dashboard, as it will allow combining sentiment analysis of data on different platforms such as Text, File, YouTube, Bluesky, and Financial news. This approach is effective due to the combination of VADER and TextBlob using an NLP engine to classify real-time data accurately and gain domain knowledge in the financial sector respectively. It has been run on five

datasets. The backend, Fast API, and the frontend, React.js, can be expanded when necessary.

The API backend written in fast API with React.js on the frontend can be expanded upon need. Each source is a distinct service and one can add more without altering the pipeline itself. This project is a good starting point in creating a complete-fledged sentiment intelligence solution that is multi-platform. It also demonstrates how to apply a natural language processing to over one third-party API in the natural world, which adds value.

REFERENCES

- [1] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, no. 1–2, pp. 1–135, 2008.
- [2] C. J. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," in Proc. 8th Int. Conf. Weblogs Social Media (ICWSM), Ann Arbor, Michigan, 2014.
- [3] S. Loria, "TextBlob: Simplified Text Processing," 2018. [Online]. Available: <https://textblob.readthedocs.io>
- [4] Y. Liu, Z. Mao, R. Geng, J. Chen, and H. Ye, "FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining," in Proc. IJCAI 2020, pp. 4513–4519.
- [5] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," Journal of Computational Science, vol. 2, no. 1, pp. 1–8, 2011.
- [6] S. Ramírez, "FastAPI: Modern, Fast Web Framework for Building APIs with Python 3.6+," 2018. [Online]. Available: <https://fastapi.tiangolo.com>
- [7] Google LLC, "YouTube Data API v3 Reference," 2024. [Online]. Available: <https://developers.google.com/youtube/v3>
- [8] AT Protocol Authors, "AT Protocol Specification — Bluesky Social," 2023. [Online]. Available: <https://atproto.com/docs>
- [9] Finnhub.io, "Finnhub Stock API — Real-Time Stock Data and Financial News," 2024. [Online]. Available: <https://finnhub.io/docs/api>
- [10] Recharts Team, "Recharts — A Composable Charting Library Built on React Components," 2021. [Online]. Available: <https://recharts.org>