Daily News Summary

Automation of News Collection, Analysis, and Forecasting

Author: Anmol Poonia

Date: January 7, 2025

Contents

1. Introduction

This project is designed as part of a data science portfolio to showcase expertise in automation, natural language processing (NLP), data visualization, and forecasting. The system fetches, analyzes, and visualizes news data from multiple sources. It provides actionable insights and trends using state-of-the-art machine learning techniques and automation tools.

1.1. Purpose

The primary goals of this project are:

- To demonstrate end-to-end data science capabilities, including data collection, processing, analysis, and reporting.
- To provide a dynamic, automated system for extracting and summarizing news insights.
- To enable sentiment forecasting and trend analysis over time.
- To showcase the use of GitHub Actions for workflow automation.

1.2. Motivation

In today's fast-paced digital world, the volume of news and information available online is overwhelming. It becomes challenging to stay updated with relevant and meaningful insights. This project aims to address this issue by automating the process of news collection, summarization, and analysis. By providing concise summaries, sentiment trends, and visualizations, it helps users make informed decisions and stay updated efficiently. The project demonstrates how data science techniques can solve real-world challenges related to information overload and time management.

2. Project Workflow

2.1. Architecture Overview

The project consists of the following components:

1. **Data Collection:** News articles are fetched from multiple sources (e.g., VentureBeat and NewsAPI).
2. **Data Processing:** Headlines are summarized and analyzed for sentiment.
3. **Data Storage:** Historical data is stored in a CSV file for trend analysis.
4. **Visualization:** Word clouds and leaderboards are generated for quick insights.
5. **Forecasting:** Sentiment trends are predicted using the Prophet library.
6. **Automation:** GitHub Actions ensures the project runs twice daily without manual intervention.

3. Components and Methodologies

3.1. Data Collection

Sources:

- **VentureBeat:** Technology-focused news articles.
- **NewsAPI:** A comprehensive API providing access to various news sources like TechCrunch and Reuters.

Implementation:

```
- Python libraries used: requests, BeautifulSoup.
- HTTP headers were added to mimic browser requests and bypass scraping restrictions.
```

3.2. Summarization and Sentiment Analysis

Summarization:

- NLP model: Facebook BART-Large-CNN.
- Purpose: To condense lengthy headlines into concise summaries.
- Mathematical Foundation: Encoder-Decoder Transformer model.

3.2.1 Mathematical Framework of BART

BART is a sequence-to-sequence model that reconstructs corrupted inputs. It uses the following:

- **Encoder:** Encodes the input sequence $X = \{x_1, x_2, \ldots, x_T\}$ into a latent representation H:

$$H = f(X; \theta)$$

- **Decoder:** Generates the output sequence $Y = \{y_1, y_2, \ldots, y_T\}$ based on H:

$$P(Y|X) = \prod_{t=1}^{T} P(y_t|H, y_{1:t-1}; \phi)$$

Sentiment Analysis:

- NLP model: DistilBERT (HuggingFace pipeline).
- Purpose: To classify news as Positive, Neutral, or Negative.
- Mathematical Foundation: Uses softmax probabilities for classification.

3.2.2 Mathematics Behind Sentiment Analysis

The model outputs logits z for each sentiment class. The probabilities are computed as:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

Where z_k is the logit for class k and $P(y = k|x)$ is the probability of class k.

3.3. Visualization

Word Cloud:

- Library: WordCloud.
- Purpose: To display the most frequent words from headlines.

Leaderboard:

- Method: Counts the most frequently occurring words in headlines.
- Output: Top 5 words with counts.

3.4. Forecasting

Prophet Library:

- Purpose: To predict future sentiment trends based on historical data.
- Mathematical Foundation: Generalized additive model (GAM) for time-series forecasting.
- Input: Historical sentiment data grouped by date.

3.4.1 Prophet Model Equations

The Prophet model predicts values $y(t)$ as:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Where:

- $g(t)$: Trend function modeling non-periodic changes.
- $s(t)$: Seasonal component modeling periodic changes.
- $h(t)$: Holiday effects.
- ε_t: Error term capturing noise.

4. Automation with GitHub Actions

Workflow Description: The workflow is defined in '.github/workflows/news_update.yml' and performs the following steps:

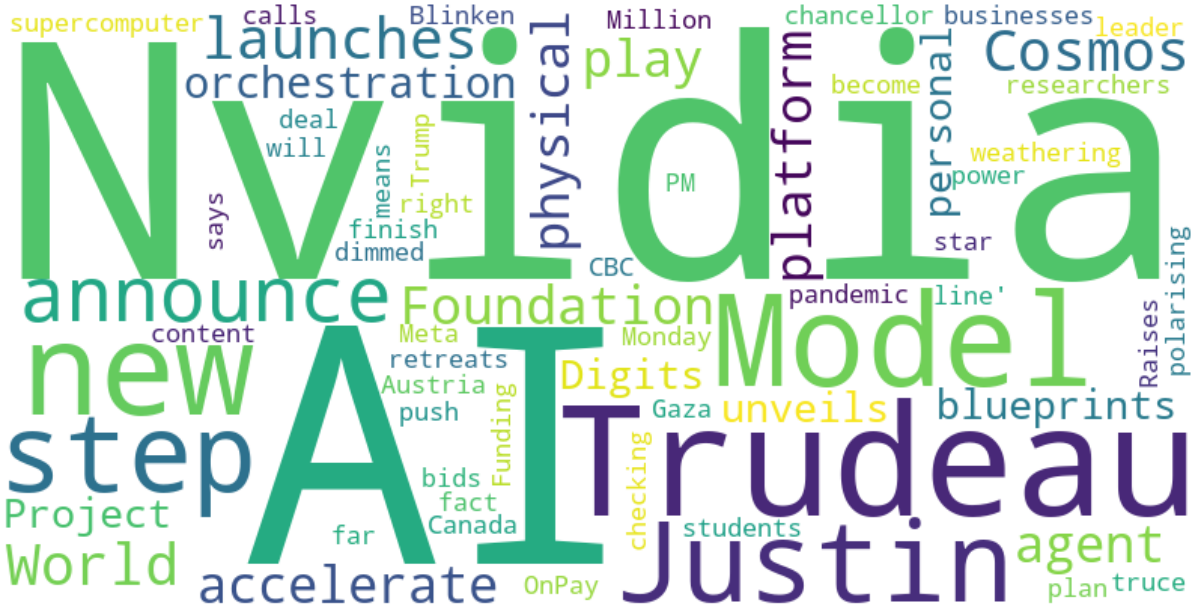
1. Clones the repository.
2. Sets up the Python environment and installs dependencies.
3. Executes the 'daily_news.py' script.
4. Commits and pushes updates to the repository.

5. Results and Outputs

5.1. Generated Files

- **news_summary.txt:** Summarized news articles.
- **word_cloud.png:** Visual representation of word frequency.
- **leaderboard.txt:** Top 5 most frequent words.
- **news_history.csv:** Historical record of fetched news.
- **sentiment_forecast.csv:** Predicted sentiment trends (available after 20 days).

5.2. Example Word Cloud



5.3. Leaderboard Example

```
Top 5 Words in News Titles:
1. AI: 15 mentions
2. Nvidia: 12 mentions
3. Justin: 8 mentions
4. Trudeau: 8 mentions
5. Platform: 7 mentions
```

6. Future Enhancements

1. Develop an interactive dashboard using Streamlit or Dash.
2. Integrate more news sources for diversified data.
3. Add email notifications for daily updates.
4. Enhance NLP models for better summarization and sentiment classification.

7. Conclusion

This project demonstrates end-to-end automation and analysis capabilities in the data science domain. By integrating state-of-the-art tools and techniques, it serves as a valuable addition to any portfolio and provides meaningful insights into news trends over time. The mathematical rigor and practical automation ensure its relevance and impact in real-world applications.