Cloud-Native Cognitive News Analysis Using PySpark and Transformers

Laith Habash And Omar Alwawi
Course: Cloud Computing

Abstract

This project presents a cloud-native pipeline that ingests, processes, analyzes, and stores news content using cognitive computing techniques. It fetches real-time news from public APIs, cleans and preprocesses text using NLP, applies sentiment/emotion analysis using transformer-based LLMs, and stores outputs in Firebase. The integration of open-source tools like PySpark (planned), Hugging Face Transformers, and Firebase supports scalable, intelligent analytics. Results are visualized and discussed, showcasing both the capabilities and limitations of the approach.

Introduction

Overview:
In an era of information overload, automated and intelligent news analysis is vital. This project addresses the challenge of deriving emotional and thematic insights from news content in real-time using modern cloud-based tools.

Objectives & Scope:

- Fetch real-time global news.
- Preprocess and clean content.
- Apply LLMs to classify emotional tone.
- Store insights using Firebase.
- Align the solution with cloud-native principles for scalability.

Background

- **PySpark:** A Python API for Apache Spark, enabling large-scale distributed data processing (planned but not implemented in this notebook).

- **Cognitive Computing:** Simulates human thought processes using NLP, ML, and pattern recognition.
- **Transformers (Hugging Face):** A powerful framework for using large pre-trained models like BERT for emotion classification.
- **Firebase:** A cloud-based backend service for storing and syncing data in real-time.

Cloud-Native Relevance:
The solution leverages modular, API-driven, scalable components such as REST APIs, pre-trained models, and Firebase storage—ideal for cloud deployment.

Methodology

1. **Data Acquisition:**
    - Use NewsAPI to fetch top 100 English-language articles via HTTP GET.
    - Extract structured fields (title, content, author, published date).
2. **Text Preprocessing:**
    - Use `nltk` to:
        - Tokenize text
        - Remove stopwords
        - Lemmatize tokens
3. **Emotion Classification:**
    - Apply a fine-tuned `distilbert-base-uncased` transformer model using `transformers` and `torch`.
    - Predict emotional labels for each article.
4. **Cloud Integration:**
    - Use `firebase-admin` to authenticate and push processed data to Firebase Realtime Database.
5. **Visualization & Debugging:**
    - Printed structured JSON examples to verify results and inspect predictions.

Results and Discussion

Outputs:

- Articles successfully fetched and cleaned.
- Emotional tones predicted (e.g., joy, sadness, anger).
- Data stored in Firebase for external access and app integration.

Visualizations:

- JSON dumps of processed records (title, emotion).
- Debug logs confirming Firebase writes.

Challenges & Solutions:

- Incomplete or null content in some articles → Resolved by filtering null entries.
- Firebase credential configuration required care.
- Tokenizer slowdowns for long texts → Handled by truncating or filtering.

Conclusion

Summary:
The project successfully demonstrated a cloud-native cognitive news analytics pipeline using open APIs, NLP, transformer-based emotion classification, and Firebase integration.

Achievements:

- Real-time data collection and processing.
- Deployment-ready architecture with Firebase.
- Insightful emotional analysis using LLMs.

Future Work:

- Integrate PySpark for distributed processing.
- Add visual dashboards (e.g., Streamlit or Firebase UI).
- Expand to multilingual news and real-time sentiment tracking.

References

- NewsAPI: `https://newsapi.org/`
- Hugging Face Transformers: `https://huggingface.co/transformers/`
- Firebase Admin SDK: `https://firebase.google.com/docs/admin/setup`
- NLTK: `https://www.nltk.org/`
- TensorFlow LLM: `https://www.tensorflow.org/text/guide/bert_preprocessing`