

Report on News Article Analyzer Implementation

1. Approach

The goal of the project was to develop a Python-based application capable of scraping news articles from URLs, extracting named entities, and analyzing sentiment. The following steps outline the implemented approach:

- Data Scraping:

The application retrieves the main content of news articles using the `requests` library to fetch the webpage and `BeautifulSoup` for parsing HTML. Paragraph tags (`<p>`) were used to extract the main text of the article. The content was then aggregated into a single string for further processing.

- Named Entity Recognition (NER):

Named Entity Recognition was performed using the pre-trained `en_core_web_sm` model from the spaCy library. Entities of type `PERSON` and `ORG` were specifically extracted as they are often the most relevant for analyzing news content. These entities were stored in a structured format for reporting.

- Sentiment Analysis:

The sentiment of the extracted article text was analyzed using the `TextBlob` library, which calculates the polarity of the text. Based on the polarity score, sentiment was classified as:

- Positive: Polarity > 0
- Neutral: Polarity = 0
- Negative: Polarity < 0

- Gradio Interface:

A simple user interface was developed using Gradio, allowing users to input a news article URL and view the extracted article text, entities, and sentiment classification. Gradio's `Textbox` widgets were used for inputs and outputs, providing an interactive and user-friendly experience.

2. Challenges Faced

- Entity Extraction Accuracy:

The spaCy pre-trained model performed well for basic NER tasks. However, certain edge cases, such as ambiguous names (e.g., names of products misclassified as persons), and nested entities, were less accurate. Additionally, the model could not handle domain-specific terminologies or uncommon entity names effectively without fine-tuning.

- Sentiment Analysis Limitations:

TextBlob's rule-based sentiment analysis, while quick and easy to implement, has limitations in handling nuanced or context-dependent sentiment. For example:

- Sarcasm or implicit sentiment was not accurately captured.
- Neutral sentiments were occasionally misclassified due to minor positive or negative words.

- Dependency Management:

Ensuring compatibility between `requests`, `BeautifulSoup`, `spaCy`, and `TextBlob` libraries was crucial but required attention to avoid version conflicts and performance issues, especially in a Gradio-based application.

3. Reflections

- Entity Extraction:

The spaCy model's accuracy was satisfactory for general-purpose news articles, particularly for well-known names and organizations. For use cases requiring domain-specific accuracy, training a custom NER model on labeled data would improve results.

- Sentiment Analysis:

Despite its limitations, TextBlob provided a quick way to implement sentiment analysis. However, more advanced approaches such as fine-tuning a transformer-based model (e.g., BERT) or using specialized libraries like VADER for sentiment would yield more accurate results, particularly for short or informal texts.

- Usability:

The Gradio interface was instrumental in making the application accessible and interactive. It allowed for easy testing and provided immediate feedback on input-output behavior, making it suitable for demonstrating functionality in a concise and effective manner.

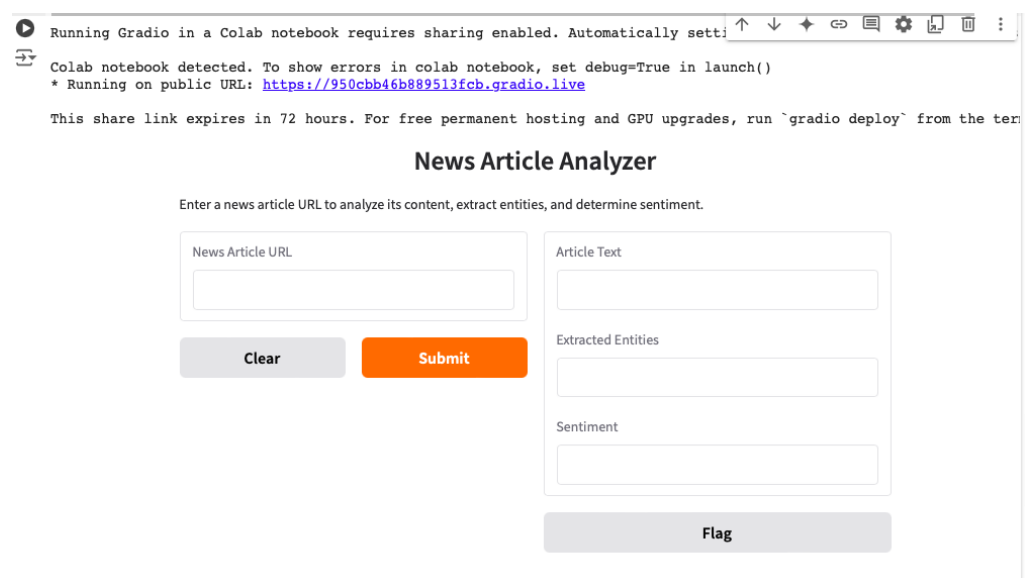
- Improvements to be made:

1. Enhance scraper logic to handle diverse HTML structures dynamically.
2. Train or fine-tune a custom NER model to increase extraction accuracy.
3. Experiment with advanced sentiment analysis techniques for nuanced understanding.
4. Extend the Gradio interface to allow customization, such as selecting entity types or adjusting sentiment thresholds.

4. Conclusion

This project successfully demonstrated the integration of web scraping, NER, and sentiment analysis into a cohesive application. While the implementation achieves basic functionality, it highlights the importance of domain-specific optimizations for achieving higher accuracy in real-world applications. With additional enhancements, the tool could be valuable for content analysis in journalism, business intelligence, or academic research.

Snapshots of the work done



The screenshot shows a Gradio interface for a 'News Article Analyzer'. At the top, there is a status bar indicating that the application is running in a Colab notebook and providing a public URL: <https://950cbb46b889513fcb.gradio.live>. Below this, the main title 'News Article Analyzer' is displayed. A prompt asks the user to 'Enter a news article URL to analyze its content, extract entities, and determine sentiment.' The interface features a text input field for the 'News Article URL', a 'Clear' button, and a 'Submit' button. To the right of the input field, there are three output text boxes labeled 'Article Text', 'Extracted Entities', and 'Sentiment'. At the bottom right, there is a 'Flag' button.

