Reflective Journal
NewsBot 2.0: AI-Driven News Intelligence System

Student: John Castor
Course: ITAI 2373 – Final Project
Date: August 2025

Introduction

This journal reflects my personal journey in building NewsBot 2.0, an AI-powered news analysis and conversational system. The project required me to design and integrate multiple natural language processing (NLP) components, create an intuitive user interface, and ensure that every part of the pipeline—from raw data ingestion to final conversational responses—worked seamlessly. I handled every stage of the work, from foundational setup to final deployment, ensuring a complete and functioning end-to-end system.

Project Kickoff and Data Exploration

The starting point was understanding the dataset, which consisted of BBC News articles. I began with exploratory data analysis (EDA) to understand distributions, missing values, and the variety of categories. I implemented a DataValidator to enforce schema consistency and renamed dataset columns for uniformity across the system. This stage was critical because any inconsistency in the data would ripple through the entire pipeline.

Text Preprocessing

Next, I developed the TextPreprocessor module to clean and prepare raw text. This included lowercasing, stopword removal, lemmatization, and language detection using langdetect. I had to ensure this preprocessing pipeline was reusable and robust, as it would be applied to all text before classification, sentiment analysis, or topic modeling. At this stage, I also integrated NLTK stopwords and handled language-specific quirks.

Feature Extraction and Classification

With the data cleaned, I moved on to implementing the FeatureExtractor using TF-IDF vectorization. I created functionality to train, save, and later reload the vectorizer, which was essential for maintaining consistency between training and prediction phases.

I built the classification module using a Naive Bayes model, trained it on the cleaned dataset, and evaluated it to achieve over 96% accuracy. I implemented methods to save and load the classifier so it could be reused without retraining every time.

Topic Modeling

For topic modeling, I implemented Latent Dirichlet Allocation (LDA) to extract themes from articles. Tuning the number of topics and ensuring they were meaningful was a challenge. I learned to interpret the results not just numerically but contextually, making the model outputs more understandable and relevant for the end user.

Sentiment Analysis and Named Entity Recognition

I built sentiment analysis using TextBlob to assign polarity scores and classify them into positive, negative, or neutral sentiments. For named entity recognition (NER), I used spaCy to identify people, organizations, dates, and other entities. This allowed NewsBot 2.0 to answer questions like “Who is mentioned in this article?” with specific and accurate responses.

Text Summarization

I implemented both extractive and transformer-based summarization approaches. The extractive model used the TextRank algorithm, while the transformer-based model leveraged Hugging Face pipelines for abstractive summaries. This gave users the choice between quick keyword-based summaries and more human-like generated summaries.

Multilingual Processing

To make the system globally relevant, I integrated language detection and translation using the Google Translate API. This allowed the system to detect non-English articles and translate them to English for consistent processing.

Conversational Interface

I built the QueryProcessor to act as the brain of the system, routing user questions to the correct NLP component. A major milestone was modifying it so that the classifier worked with feature extraction for category prediction, solving the “expected 2D array” error. This made it possible for the conversational interface to respond accurately to a range of queries, from “What is the sentiment?” to “Summarize this article.”

Deployment and Interactive Interface

For deployment, I used FastAPI and Gradio to provide both API endpoints and a user-friendly web interface. This allowed users to paste article text and ask follow-up questions in a natural conversational style. Debugging API errors and resolving compatibility issues with pydantic was one of the final technical challenges.

Model and Artifact Management

I implemented the ability to save and load the TF-IDF vectorizer and classifier into /data/models/ so they could be versioned and reused without retraining. This was essential for consistency and reproducibility.

Lessons Learned

The biggest lesson was the importance of **modular architecture**. By keeping every NLP component separate but connected, I could test, debug, and improve each part without breaking the others. I also learned how critical **version control** and **artifact management** are for complex AI systems.

I gained hands-on experience in integrating multiple NLP libraries, handling pipeline dependencies, and troubleshooting integration errors. This process reinforced my understanding of how professional AI projects are structured and deployed.

Conclusion

From raw dataset to a fully deployed conversational AI system, I was responsible for every stage of NewsBot 2.0's creation. This project gave me the skills to handle both the technical and structural challenges of an end-to-end AI application. I now have a complete, functioning portfolio piece that demonstrates expertise in natural language processing, machine learning, and AI deployment.