Technical Documentation
NewsBot 2.0: AI-Driven News Intelligence System

Student Team: John, Dylan, Milagros, Ola
Course: ITAI 2373 – Final Project
Date: August 2025

1. Introduction

This technical documentation describes the design, development, and deployment of *NewsBot 2.0*, an AI-powered news intelligence system. The system integrates multiple natural language processing (NLP) modules—classification, sentiment analysis, named entity recognition, topic modeling, summarization, and multilingual translation—into a unified framework that can be accessed via a conversational interface. The purpose of this project is to create a tool capable of ingesting raw news articles, processing them through a structured AI pipeline, and returning meaningful, actionable insights for the user.

2. System Overview

NewsBot 2.0 is organized into modular components, each responsible for a specific function in the news analysis workflow:

- **Data Processing Module** – Handles preprocessing of raw article text, including tokenization, stopword removal, lemmatization, and language detection.
- **Feature Extraction Module** – Uses TF-IDF vectorization to convert text into numerical features suitable for machine learning models.
- **Classification Module** – Employs a Naive Bayes classifier to categorize articles into five categories: business, entertainment, politics, sport, and tech.
- **Sentiment Analysis Module** – Determines the polarity and sentiment label (positive, negative, neutral) of the article's content.
- **Named Entity Recognition (NER) Module** – Identifies and labels entities such as people, organizations, locations, dates, and monetary amounts.
- **Topic Modeling Module** – Uses Latent Dirichlet Allocation (LDA) to assign main topics and display associated keywords.
- **Summarization Module** – Provides both extractive and transformer-based abstractive summaries.
- **Multilingual Support Module** – Detects article language and translates non-English content into English.

- **Conversational Interface Module** – Allows users to interact with the system through natural language queries, retrieving results from relevant modules.

3. Development Environment

The system was developed in **Python 3.11** using Google Colab for iterative development and testing. Key libraries include:

- **NLTK** – Stopword removal and text preprocessing
- **spaCy** – Lemmatization and NER
- **langdetect** – Language detection
- **scikit-learn** – Classification and feature extraction
- **pyLDAvis** – Topic modeling visualization
- **transformers** – Abstractive text summarization
- **sumy** – Extractive text summarization
- **googletrans** – Translation
- **gradio** – Web-based conversational interface
- **FastAPI** – Backend API for deployment

4. Data Sources

The dataset used for training and testing was the *BBC News Dataset*, which contains labeled news articles in five categories. The dataset was divided into training and test sets to evaluate classifier performance.

5. Implementation Workflow

1. **Data Exploration** – Load and inspect dataset, check for missing values, and analyze category distribution.
2. **Preprocessing** – Standardize text format, remove unwanted characters, apply tokenization, lemmatization, and stopword removal.
3. **Feature Extraction** – Convert cleaned text to TF-IDF feature vectors.
4. **Model Training** – Train Naive Bayes classifier and evaluate accuracy.
5. **Topic Modeling** – Apply LDA to identify and visualize topics.
6. **Sentiment Analysis** – Assign sentiment polarity and label.

7. **NER** – Extract named entities with associated labels.
8. **Summarization** – Generate summaries using extractive and abstractive methods.
9. **Translation** – Detect non-English text and translate to English.
10. **Conversational Interface** – Implement query processor to route user requests to the appropriate module.

6. Results

- **Classifier Accuracy:** 96.64% on test set.
- **Topic Modeling:** Effective separation of key topics with relevant keywords.
- **Sentiment Analysis:** Correctly labeled sample articles as positive, negative, or neutral.
- **Summarization:** Produced concise summaries with key facts preserved.
- **NER:** Successfully extracted named entities with correct labels.

7. Conclusion

NewsBot 2.0 successfully integrates multiple NLP techniques into a single cohesive tool for news article analysis. Its modular design allows for future improvements such as adding more categories, expanding multilingual capabilities, and integrating with real-time news APIs. The combination of high accuracy in classification and flexibility in query handling makes it a valuable proof-of-concept for real-world news intelligence applications.

References

- Bird, Steven, Edward Loper, and Ewan Klein. *Natural Language Processing with Python.* O'Reilly Media, 2009.
- Vaswani, Ashish, et al. "Attention Is All You Need." *Advances in Neural Information Processing Systems*, 2017.
- Pedregosa, Fabian, et al. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830.