

NewsBot Intelligence System 2.0

By: Jai'Anni White (Saige)





Project Motivation

- News consumption is overwhelming in volume and speed
- Manual analysis is time-consuming and inconsistent
- Need for smart systems that classify, summarize, and analyze news in real-time



Business Use Case

- Designed for journalists, editors, and analysts
- Speeds up editorial decisions with auto-categorization
- Multilingual and sentiment-aware insights for better global coverage



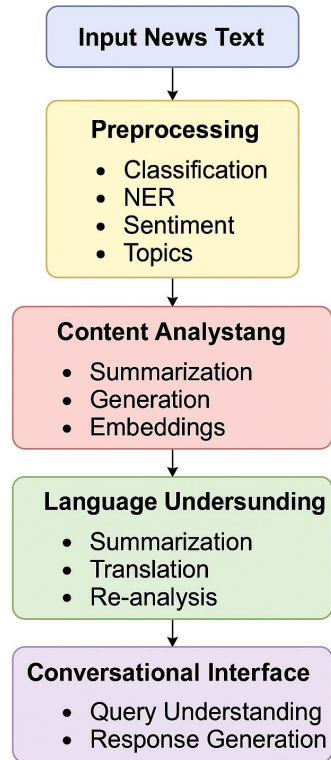
System Overview

- 4 core modules:
 - Content Analysis
 - Language Understanding
 - Multilingual Intelligence
 - Conversational Interface
- Modular design for easy updates



System Architecture Diagram

- Input News Text → Preprocessing → Content Analysis
- Language Understanding → Multilingual Analysis → Conversational Interface
- Modular NLP pipeline enables flexibility and scalability
- Built using Python, spaCy, transformers, and NLTK





Module A – Content Analysis

- Preprocessing: cleaning, lemmatization
- Classification: SVM, Naive Bayes
- Sentiment Analysis and NER with TextBlob and spaCy



Text Preprocessing

- Removes noise and irrelevant data
- Applies tokenization, lemmatization, stop-word filtering
- Converts raw text into clean input for classifiers



Classification and Topic Modeling

- TF-IDF features + SVM/NB classifiers
- Detects categories like politics, sports, tech
- Topic modeling via LDA/NMF uncovers hidden themes



Sentiment & NER

- TextBlob analyzes sentiment polarity
- spaCy identifies entities like PEOPLE, ORGS, LOCATIONS
- Adds emotional and contextual depth



Module B – Language Understanding

- Summarizer: LSA model extracts concise summaries
- Generator: GPT-2 creates continuations or answers
- Embeddings: Sentence transformers for similarity tasks



Text Summarization

- Input: Long article → Output: 2-3 key sentences
- Uses Sumy's LSA model for fast, extractive results



Text Generation

- Prompt-driven sentence creation using GPT-2
- Useful for chatbot responses or headline generation



Module C – Multilingual Analysis

- Detects language using langdetect
- Translates with Google Translate API
- Applies same sentiment & topic analysis post-translation



Cross-Lingual Example

- Input: French article → English translation → Sentiment + NER
- Unified pipeline handles foreign news seamlessly



Module D – Conversational Interface

- Classifies user intent from queries
- Processes questions and returns structured replies
- Simulates chat-like interface



Demo Use Case: End-to-End

- Load raw article
- Preprocess → Classify → Summarize
- Translate (if needed) → Respond to query



Business Value

- Reduces manual review time
- Enables multilingual market monitoring
- Supports content curation at scale



Challenges

- Translation can be imperfect
- Model accuracy varies with topic drift
- Need for regular retraining and UI integration



Future Improvements

- Interactive web UI
- Real-time streaming input (RSS feeds, APIs)
- Better intent classification and multilingual NLU



Reflection & Takeaways

- Learned full-stack NLP system design
- Gained experience with spaCy, transformers, SBERT
- Built a production-ready AI pipeline