NewsBot 2.0 – Technical Documentation

Architecture Overview

NewsBot 2.0 is built on a modular architecture with the following core components:

1. Data Ingestion Layer: Loads and splits the BBC News dataset into training, test, and sample sets.

2. Preprocessing Module: Cleans and normalizes raw text, performs tokenization, stop word removal, and lemmatization.

3. Analysis Engine: Includes topic modeling with LDA, sentiment analysis, clustering, and semantic search.

4. Language Module: Supports summarization, translation, and cross-lingual NLP features.

5. Conversation Layer: Classifies user intent, processes natural language queries, manages context, and generates responses.

6. Output Layer: Visualizes topics, clusters, and results; provides feedback for users and developers.

API Reference

- preprocess_text(text): Clean and normalize input text.

- perform_topic_modeling(corpus, n_topics): Apply LDA to extract topic distributions.

- generate_summary(text): Return a concise summary using TextRank or similar method.

- detect_language(text): Identify the language of the given text.

- translate_text(text, target_lang): Translate text to the target language.

- classify_intent(query): Classify the type of user query.

- generate_response(query): Return a generated answer based on query and context.

Installation Guide

Step 1: Clone the repository

git clone https://github.com/yourusername/ITAI2373-NewsBot-Final.git

Step 2: Set up a virtual environment (optional but recommended)

python -m venv venv

source venv/bin/activate (Linux/macOS)

venv\Scripts\activate (Windows)

Step 3: Install required dependencies

pip install -r requirements.txt

Step 4: Run the main pipeline or notebook

jupyter notebook notebooks/Final_Pipeline.ipynb

Configuration Manual

All major settings are defined in the `config/` folder.

- config/settings.yaml: Define parameters like number of topics, clustering thresholds, supported languages.

- config/paths.yaml: Set paths for datasets, model checkpoints, and output directories.

- utils/logger.py: Configure custom logging levels and formats.

You can also adjust the `TopicModeler` or `Summarizer` classes in `src/analysis/` or `src/language_models/` to experiment with alternate models or thresholds