KeyInsights: Keyword Extraction using NLP

Sourav
E23CSEU1865
Department of Computer Science
Bennett University, India

Prakhar Swaroop
E23CSEU1862
Department of Computer Science
Bennett University, India

Shreyansh
E23CSEU1869
Department of Computer Science
Bennett University, India

***Abstract*—The exponential growth of unstructured text data necessitates powerful tools for rapid analysis and insight extraction. This project presents KeyInsights, a comprehensive, interactive web application built with Streamlit, designed to perform a suite of advanced NLP tasks. The platform integrates multiple state-of-the-art models to offer three distinct analysis modes: Single Document Analysis, Comparative Analysis, and Corpus Trend Analysis.**

Key functionalities include keyword extraction (using KeyBERT), document summarization and sentiment analysis (using Hugging Face Transformers), topic modeling (using BERTopic), and knowledge graph generation (using spaCy and NetworkX). The system is designed for flexibility, supporting multilingual models, various file inputs (PDF, DOCX, TXT), and rich, interactive visualizations using Plotly and PyVis. The platform serves as an all-in-one solution for users needing to quickly understand, compare, and track themes within textual data.

***Index Terms*—NLP, KeyBERT, BERTopic, Keyword Extraction, Text Summarization, Sentiment Analysis, Knowledge Graph, Streamlit**

I. Conceptual Background & Literature Review

The KeyInsights platform is built upon several foundational concepts in modern Natural Language Processing.

A. Transformer-Based Embeddings

Unlike classical methods (e.g., TF-IDF), transformer models like BERT create deep contextual embeddings. The system leverages sentence-transformer models (e.g., all-MiniLM-L6-v2, paraphrase-multilingual-MiniLM-L12-v2) as the backbone for high-quality text understanding.

B. Keyword Extraction (KeyBERT)

The project uses KeyBERT, a technique that leverages BERT embeddings and cosine similarity to find keywords and phrases that are most representative of a document. It also implements Maximal Marginal Relevance (MMR) to enhance keyword diversity, balancing relevance with redundancy.

C. Topic Modeling (BERTopic)

For corpus analysis, BERTopic is used. This model outperforms traditional methods (like LDA) by clustering document embeddings. It enables coherent topic discovery from text without extensive preprocessing.

D. Abstractive & Extractive Summarization

The summarization module uses Hugging Face's pipeline, capable of loading models like `sshleifer/distilbart-cnn-12-6`. This enables abstractive summarization (generating new text), and as a fallback, extractive summarization (selecting key sentences).

E. Named Entity Recognition (NER)

spaCy is used to perform NER, identifying and categorizing entities such as persons, organizations, and locations. These entities also form the nodes used in the knowledge graph.

II. Data Input & Preprocessing

The platform is designed to handle user-provided data rather than a fixed dataset.

A. Data Sources

Users can provide data by either pasting raw text or uploading files. The system supports:

- `.txt` files (plain text)
- `.pdf` files via PyPDF2
- `.docx` files via python-docx

B. Preprocessing Steps

Text Extraction: Handled through `extract_text_from_file` depending on file type.

Language Detection: The function `detect_language` automatically identifies the language using `langdetect`.

Text Chunking: For large documents, the `chunk_text` function divides text into overlapping chunks to avoid transformer token limits.

Date Parsing: For trend mode, `parse_filenames_for_dates` uses regex to detect and sort documents chronologically (e.g., filenames with patterns like YYYY-MM-DD).

III. Methodology & System Architecture

The system functions as a modular Streamlit dashboard integrating an NLP processing pipeline.

A. Core Framework

Streamlit is used for the frontend, sidebar controls, and session state management.

B. Model Management

The `caching.py` module employs `@st.cache_resource` to load models only once:

- KeyBERT model
- BERTopic model
- spaCy NER model
- Hugging Face pipelines for summarization and sentiment

C. NLP Processing

The `processing.py` module contains classes for:

- Keyword extraction
- Sentiment analysis
- Abstractive summarization
- NER detection
- Knowledge graph generation
- Corpus comparison using log-likelihood keyness

D. Visualizations

- Plotly for interactive bar charts, line charts, and pie charts
- WordCloud for keyword clouds
- PyVis for interactive knowledge graph visualization

E. Reporting

The **PDFReportGenerator** (in `pdf_generator.py`) creates downloadable PDF reports using ReportLab.

IV. Core Features & Functionality

The dashboard supports three modes of analysis:

A. Single Document Analysis

Outputs include:

- AI-generated summary
- Keyword bar chart + WordCloud
- Sentiment distribution (Positive/Neutral/Negative)
- Named entities extracted using spaCy
- Knowledge graph of entity relationships

B. Comparative Analysis

Given two documents, the system provides:

- Keyness analysis via log-likelihood scoring
- Side-by-side keyword plots
- Sentiment comparison and tables

C. Trend Analysis (Corpus)

For 3 or more documents:

- Sentiment trends over time via Plotly line charts
- Keyword evolution trends
- Topic modeling results via BERTopic, including topic-level word clouds

V. Discussion

This project successfully integrates multiple state-of-the-art NLP models into a single, cohesive, interactive tool. Streamlit allows rapid development of a rich and intuitive interface, eliminating the need for specialized frontend development.

The modular architecture enables scalability and maintainability—new models or processing modules can be added easily. The system's flexibility, including multilingual model support, adjustable keyword settings, and export options (PDF, CSV), makes it useful for researchers, data analysts, and students.

VI. Conclusion

The KeyInsights platform demonstrates the successful development of a powerful, interactive NLP tool capable of performing keyword extraction, summarization, sentiment analysis, knowledge graph generation, and topic modeling. By combining KeyBERT, BERTopic, spaCy, and Hugging Face transformers, the system provides a comprehensive toolkit for analyzing and understanding textual data. It serves as a practical example of end-to-end NLP system design using modern transformer-based pipelines.

References

[1] Streamlit, "Web Application Framework for Python," 2019.
[2] Hugging Face Transformers, "State-of-the-art NLP Models," 2020.
[3] M. Grootendorst, "KeyBERT: Keyword Extraction with BERT," 2020.
[4] M. Grootendorst, "BERTopic: Transformer-based Topic Modeling," 2021.
[5] M. Honnibal et al., "spaCy: Natural Language Processing in Python," 2020.
[6] ReportLab, "PDF Generation Toolkit for Python," 2022.