Semantic Search in articles using NLP

1. Introduction

1.1. Problem Statement

In the era of information overload, finding relevant content from a large set of articles is crucial for many applications, such as content recommendation, summarization, and information retrieval. The task involves identifying hot keywords and searching for specific words within English articles. The goal is to optimize the time it takes to extract relevant words and make the search process more efficient using natural language processing (NLP) techniques.

1.2. Objectives

- Develop a pipeline to efficiently search for specific words in English articles.
- Extract hot keywords (frequently occurring and contextually important terms) from the articles.
- Preprocess the data to make it ready for NLP tasks.
- Optimize the extraction process for speed without compromising the quality of the results.

2. Data Processing

The first step in this pipeline involves processing the raw article data to make it usable for NLP tasks:

Text Cleaning: Remove any unwanted characters, punctuation, HTML tags, or special symbols, lowercasing.

3. Sentence Embedding and Keyword Extraction

After preprocessing, we move on to transforming the text into sentence embeddings and extracting keywords:

Sentence Embeddings: Use models like **BERT** or **Sentence-BERT** to convert sentences or articles into embeddings (vectors) that capture their semantic meaning. This is useful for finding similar sentences or topics within articles

Keyword Extraction: using advanced methods like RAKE (Rapid Automatic Keyword Extraction) can be applied to extract the most relevant words that represent the article's core content.

Optimization: For large datasets, optimization techniques like **Approximate Nearest Neighbor Search (ANN)** using libraries like **FAISS** can be used to speed up the search process by reducing the time complexity of semantic searches.

4. Tools Used

The following tools were used to build the pipeline:

- Python
- NLTK
- Sentence-Transformers
- rake

5. External Resources

- Hugging Face Models
- NLTK

6. Project Learnings

6.1.Challenges

- Data Quality: Articles may contain noisy text, irrelevant content, or inconsistent formats that can affect the accuracy of keyword extraction.
- Performance Optimization: As the dataset grows, the time required for semantic search and keyword extraction may increase. Efficient algorithms (like FAISS or Annoy) need to be employed to reduce latency.

7. Conclusion

This pipeline leverages NLP techniques for semantic search and keyword extraction from articles. It includes preprocessing steps like tokenization and stopword removal, followed by sentence embedding and keyword extraction. The tools used include NLTK, scikit-learn,

Sentence-BERT, and FAISS for performance optimization. The challenges primarily involve handling large datasets, ensuring the accuracy of keyword extraction, and optimizing search times.