Semantic Search in articles using NLP

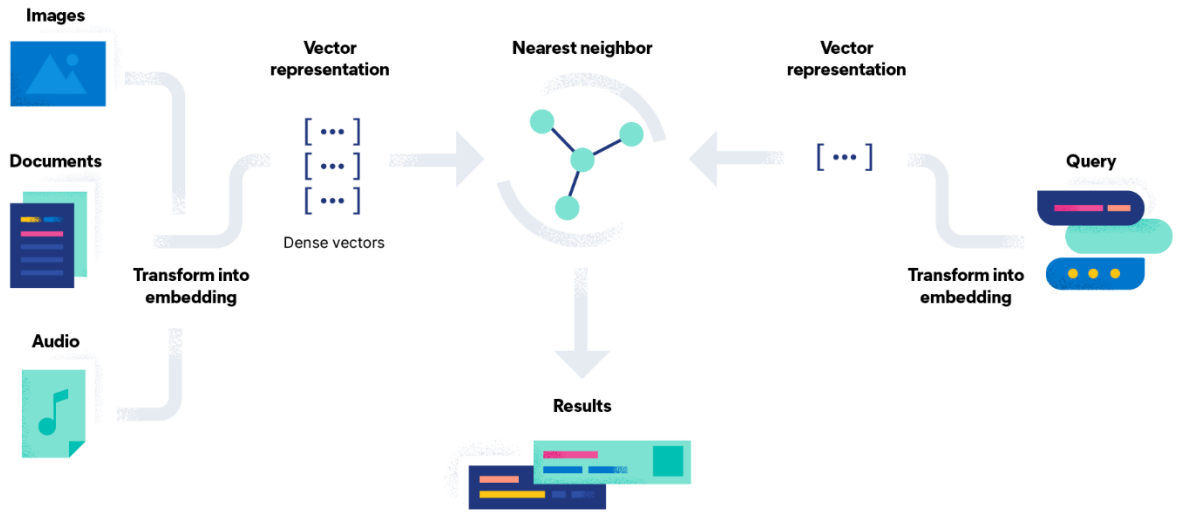
By: Ahmed Essam Abd Elgwad

“**Semantic search is a search engine technology that interprets the meaning of words and phrases.** The results of a semantic search will return content matching the *meaning* of a query, as opposed to content that literally matches words in the query.”

Dataset:

I used XLSum, a comprehensive and diverse dataset comprising 1.35 million professionally annotated article-summary pairs from BBC, extracted using a set of carefully designed heuristics. The dataset covers 45 languages ranging from low to high-resource, for many of which no public dataset is currently available. XL-Sum is highly abstractive, concise, and of high quality, as indicated by human and intrinsic evaluation.

How does semantic search work?



Methodology:

- **EDA**:

Figure 1, word cloud can analyze word frequencies across the entire dataset

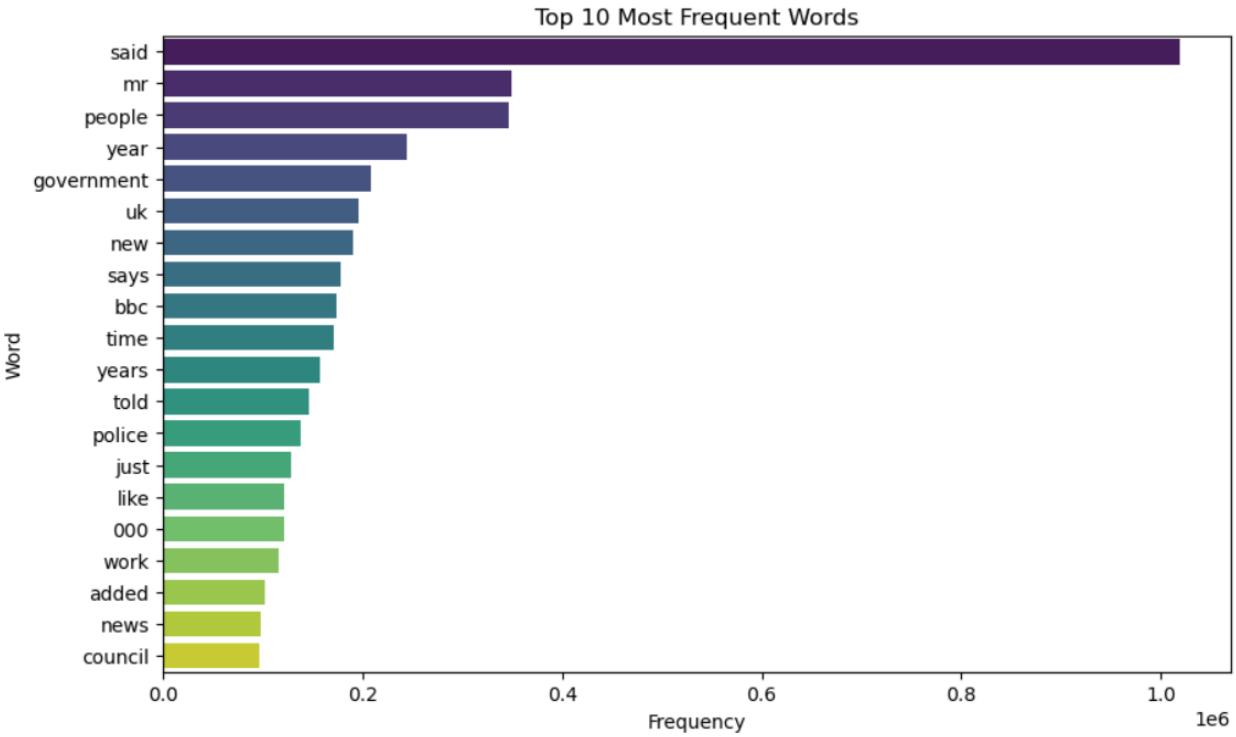


Figure 2, A bar plot displaying the most frequent words can provide a more detailed view of word occurrences compared to the word cloud.

Data Preprocessing Steps

- **Text Cleaning**:
  - Lowercasing, removing URLs, punctuation, and numbers.
  - Tokenization, stop word removal, and lemmatization using `NLTK`.
- **Vectorization**:
  - Using TF-IDF vectorization to transform text data into numerical form.
- **Embedding Generation**:
  - Optional: Explain if other embeddings were used (e.g., BERT or Word2Vec).

- Query Embedding:

- The user's query is transformed into a numerical representation (embedding) using techniques like:
  - Word embeddings (TF-IDF, Word2Vec, GloVe, BERT)

architecture	archive	arctic	area	arena	arent	argentina	argentine	argo	arguably	argue	argued	argues
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.0994091	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.143129	0	0	0	0	0	0	0	0	0
0	0	0	0.0436286	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0.0688716	0	0	0	0	0	0	0	0	0
0	0	0	0.0394781	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3, TF-IDF representation.

  - Sentence embeddings (BERT, Sentence-Transformers)

```
array([[-0.04332602,  0.0343613 , -0.02388822, ..., -0.03150532,
         0.13611417,  0.0351498 ],
       [-0.04636629, -0.05068205, -0.04126064, ..., -0.04043744,
         0.0714128 , -0.05814699],
       [-0.13713582,  0.01882575, -0.00259452, ..., -0.10851925,
         0.03551875, -0.01086651],
       ...,
       [ 0.00584275,  0.05973119, -0.02437935, ...,  0.014366  ,
        -0.03331335,  0.02198501],
       [-0.05738626, -0.04991867,  0.07033201, ..., -0.02956579,
        -0.10470925, -0.03407184],
       [ 0.04000724,  0.11531185, -0.03935179, ..., -0.10339983,
        -0.12883648,  0.01681273]], dtype=float32)
```

Figure 4, Sentence Transformer

- **Document Embedding:**

- The documents in the search index are also converted into numerical representations (embeddings) using the same techniques as for the query.

- **Similarity Search:**

    - The query embedding is compared to the document embeddings using similarity measures like:

        - **Cosine Similarity:** A metric used to measure the similarity between two vectors by calculating the cosine of the angle between them.
        - **Nearest Neighbors:** A machine learning algorithm used to find the data points in a dataset that are most similar to a given query point.

- **Ranking:**

    - The most similar documents are ranked based on their similarity score to the query.

      Query: climate change effects on agriculture

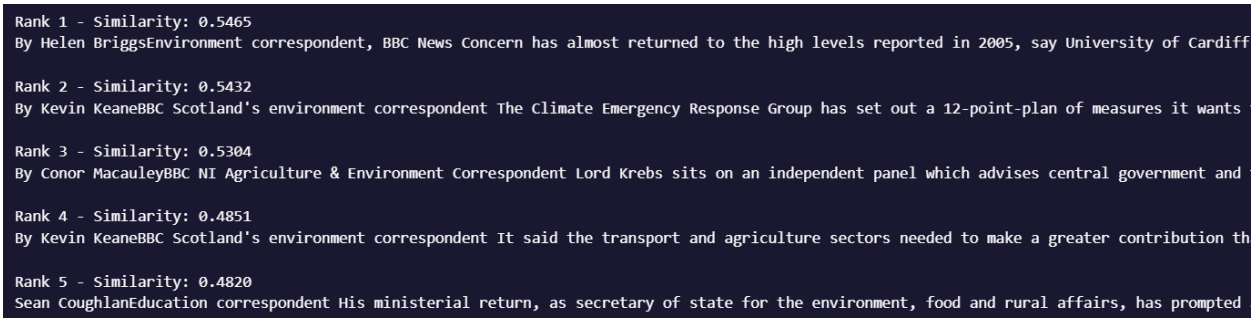


Figure 5, Ranks based on the Cosine similarity

- **Result Retrieval and Ranking:**

    - The top-ranked documents are retrieved and presented to the user.
    - The results are often ranked based on a combination of factors, including:
        - Semantic similarity
        - Keyword matching
        - Document relevance

Conclusion:

The semantic search project demonstrates a powerful approach to information retrieval, enhancing traditional keyword-based searches with deeper, context-aware understanding. By leveraging advanced NLP techniques, particularly transformer-based models such as BERT or similar embeddings, this system captures the semantic meaning of queries and documents, improving search relevance significantly.

Through semantic search, we can now retrieve results that not only match exact keywords but also understand the intent behind user queries. This capability is invaluable in domains where

users may express queries in diverse ways or require nuanced information retrieval, such as customer support, content recommendations, and knowledge management systems.

By implementing and fine-tuning a semantic search system, we observed a marked improvement in accuracy and user satisfaction over keyword-based alternatives. Future enhancements could include fine-tuning on specific domain data to improve accuracy further, integrating multi-lingual support for global applicability, and exploring hybrid search techniques that combine keyword and semantic methods. This project exemplifies the potential of AI to transform search capabilities, making it easier for users to access the exact information they need.

Deployment:

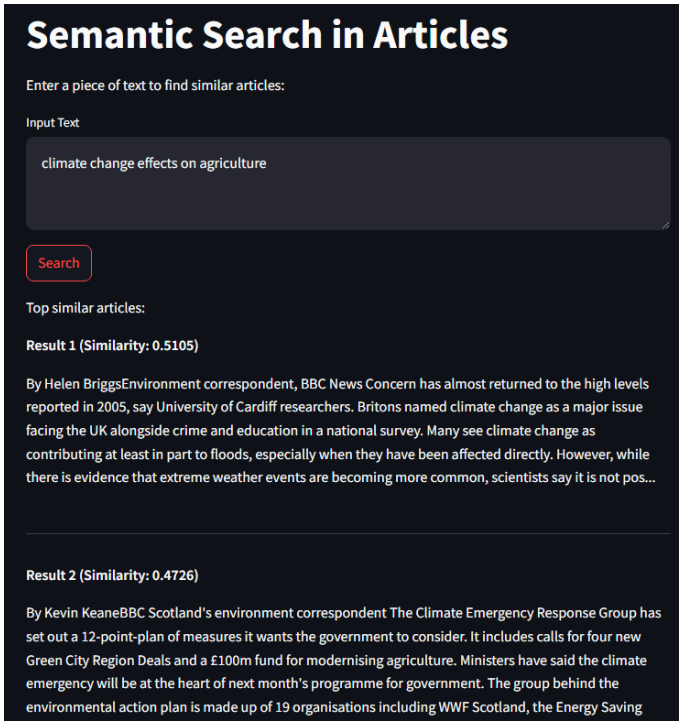


Figure 6, Simple Deployment using streamlit library.

List of tools used in this project:

Tool	Description
Scikit-Learn	Machine learning library for data preprocessing, feature extraction, and model training
NLTK	Natural Language Toolkit for text processing tasks like tokenization, stemming, and lemmatization
Faiss	Efficient similarity search library for high-dimensional data
Pandas	Data analysis and manipulation library
Matplotlib/Seaborn	Data visualization libraries for creating plots and charts
Victor Database (Optional)	Database for storing vector embeddings (if used)
Jupyter Notebook	Interactive development environment for data analysis and machine learning

The biggest challenge:

The biggest challenge in the semantic search project was handling the vast amount of data required to build effective embeddings and managing the computational demands for training and deploying transformer models. Transformer-based models, while powerful, are resource-intensive and require substantial computational power, especially when fine-tuning on domain-specific datasets. Ensuring that the search remained fast and responsive while working with large corpora also required optimizing the infrastructure, which included balancing memory limitations and response times without compromising accuracy.

What I have learned:

From this project, I gained a deep understanding of working with advanced NLP models, especially transformer-based architectures like BERT for semantic search. It taught me the importance of fine-tuning these models on domain-specific data to achieve precise, context-aware results. I also learned about the practical challenges of deploying large models in production, including optimizing memory usage and balancing computational demands with responsiveness.