**Document Similarity**

**Final Year**

in

**Artificial Intelligence and Data Science**

By

**Bhaumik S. Birje 06**

**Sahil S. Chavan   08**

**Rishikesh M. Dharane 14**

**Yash B. Dhole 15**

Supervisor

**Prof. Depali S. Chavan**



**Department of**

**Artificial Intelligence and Data Science**

**DATTA MEGHE COLLEGE OF ENGINEERING, AIROLI, NAVI MUMBAI - 400 708**

**University of Mumbai**

**(AY 2023-24)**

**Abstract**

This project focuses on detecting document similarity using Natural Language Processing (NLP) and the BERT model. By leveraging pre-trained transformer-based models, the system compares documents to compute similarity scores. The project includes uploading documents in various formats such as text, PDF, and Excel to identify similarities using cosine similarity on BERT embeddings. The system allows users to assess how closely related two documents are and provides results in percentage format. This report outlines the methodology, system architecture, and findings of the project.

***Keywords***- NLP, BERT, Document Similarity, Transformer Models, Cosine Similarity, Embeddings, Preprocessing, Text Comparison

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Name** | **Page**  **No.** |
| 1 | Similarity of words | 2 |
| 2 | Preprocessing Workflow | 4 |
| 3 | BERT Tokenization Process | 5 |
| 4 | Cosine Similarity Formula | 5 |
| 5 | Architecture of the Proposed System | 6 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name** | **Page**  **No.** |
| 1 | Literature Survey | 03 |
| 2 | Document Similarity Results for Test Data | 7 |
| 3 | Comparison of Various Algorithms for Document Similarity | 7 |
| 4 | Performance Evaluation Metrics | 8 |

**CHAPTER 1**

**Introduction**

Document similarity detection plays a crucial role in several applications like plagiarism detection, recommendation systems, and content duplication checks. Traditional methods, which relied heavily on exact keyword matching or bag-of-words approaches, failed to capture semantic meaning. With the advent of deep learning and NLP models like BERT, it's now possible to capture context and semantics at a much deeper level.

This project uses the BERT (Bidirectional Encoder Representations from Transformers) model to compute embeddings for each document and then uses cosine similarity to compare them. The user can upload documents in various formats (text, PDF, Excel), and the system outputs the similarity percentage between them.

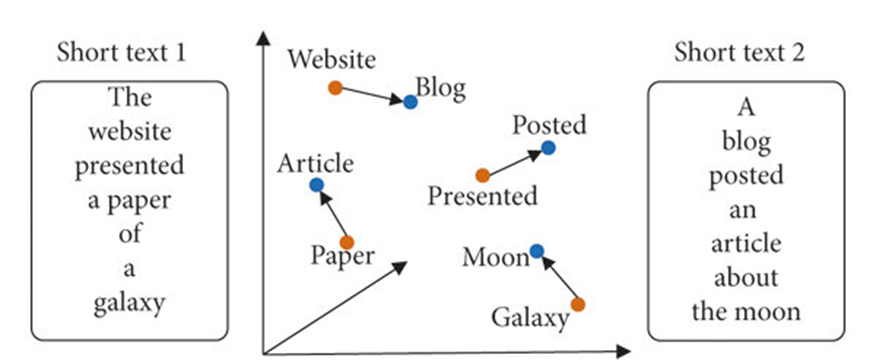


Figure 1

# **Literature Survey**

**2.1 Current Literature Survey**

Table 2.1 Explains the literature survey and key findings

|  |  |  |  |
| --- | --- | --- | --- |
| **SR NO.** | **Author Name and Year Of publication** | **Title** | **Key Finding** |
| **1.** | Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova,2018 | BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding | Introduced BERT, which improved contextual understanding and significantly enhanced NLP tasks, including document similarity. |
| **2.** | Nils Reimers, Iryna Gurevych,2019 | Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks | Improved document comparison by creating semantically meaningful sentence embeddings, reducing computational time for similarity calculations. |
| **3.** | Xinyue Zhang, Hao Wu, Zhonghai Wu, Zhenjie Ren,2021 | Using BERT Embeddings and Cosine Similarity for Document Comparison | Demonstrated that cosine similarity applied on BERT embeddings provided high accuracy for detecting similarity between documents.. |

**CHAPTER 2**

**Proposed System**

The proposed system for document similarity detection follows a streamlined architecture that involves preprocessing, embedding generation, similarity calculation, and results presentation. The system's key strength lies in its ability to handle multiple file formats and use BERT to create rich, context-aware embeddings for accurate similarity computation.

**3.1 Preprocessing**

The system supports user input in multiple formats, including text, PDF, and Excel. Upon uploading, the text is extracted and preprocessed. Preprocessing includes:

* **Text extraction**: Reading the content of PDFs, Excel sheets, and text files.
* **Cleaning**: Removing special characters, punctuation, stopwords, and numbers.
* **Tokenization**: Splitting the text into smaller units (tokens) using the BERT tokenizer.

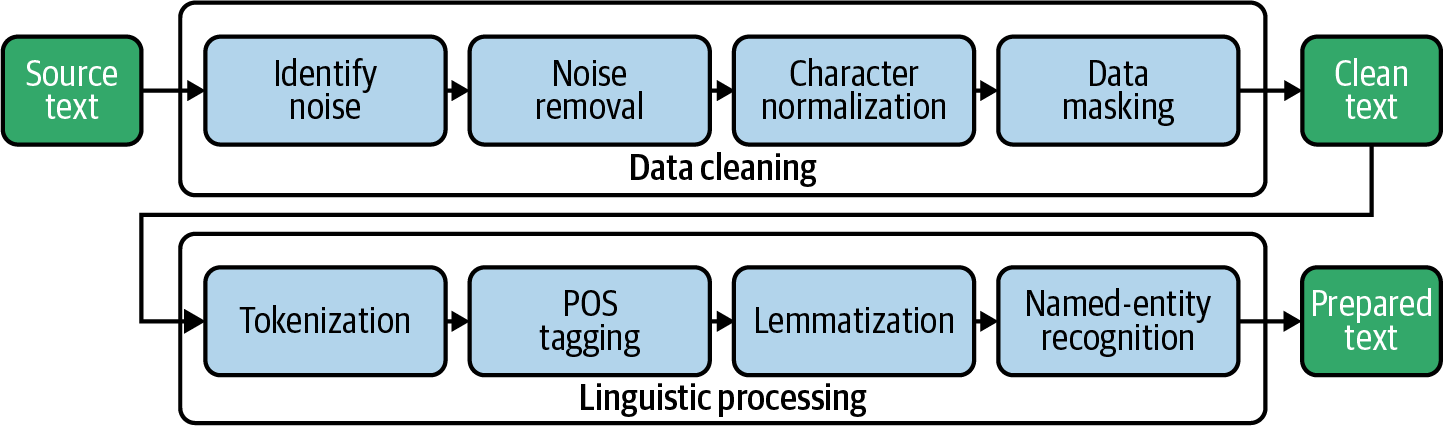


Figure 2

**3.2 Feature Extraction**

After preprocessing, BERT is used to generate embeddings for each document. The BERT model captures the semantic structure of each document by generating contextual embeddings, which are then pooled to create a single vector for each document.

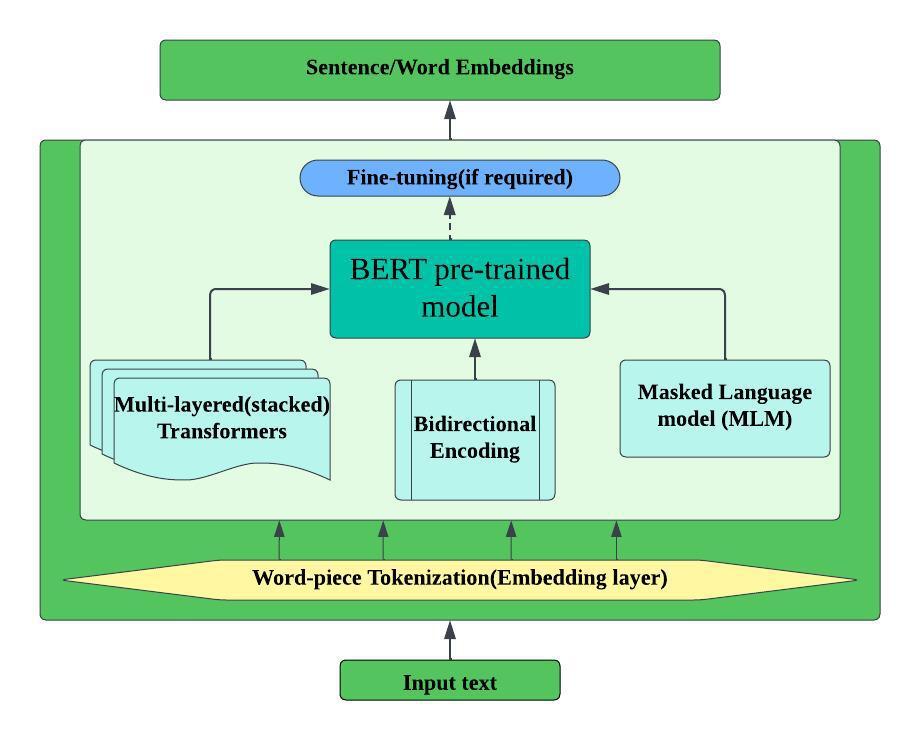
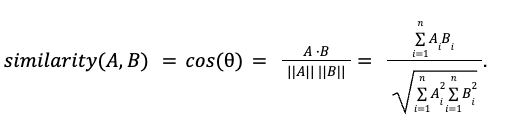


Figure 3

**3.3 Similarity Calculation**

Once the embeddings are generated, cosine similarity is applied to compute the similarity score between the two documents. The cosine similarity metric measures the angle between the two document vectors, where a smaller angle indicates greater similarity.



**3.4 Thresholding and Results**

The similarity score is converted to a percentage, and results are displayed to the user in an intuitive manner. The user can interpret how closely related the two documents are based on this percentage

**3.5 System Architecture**

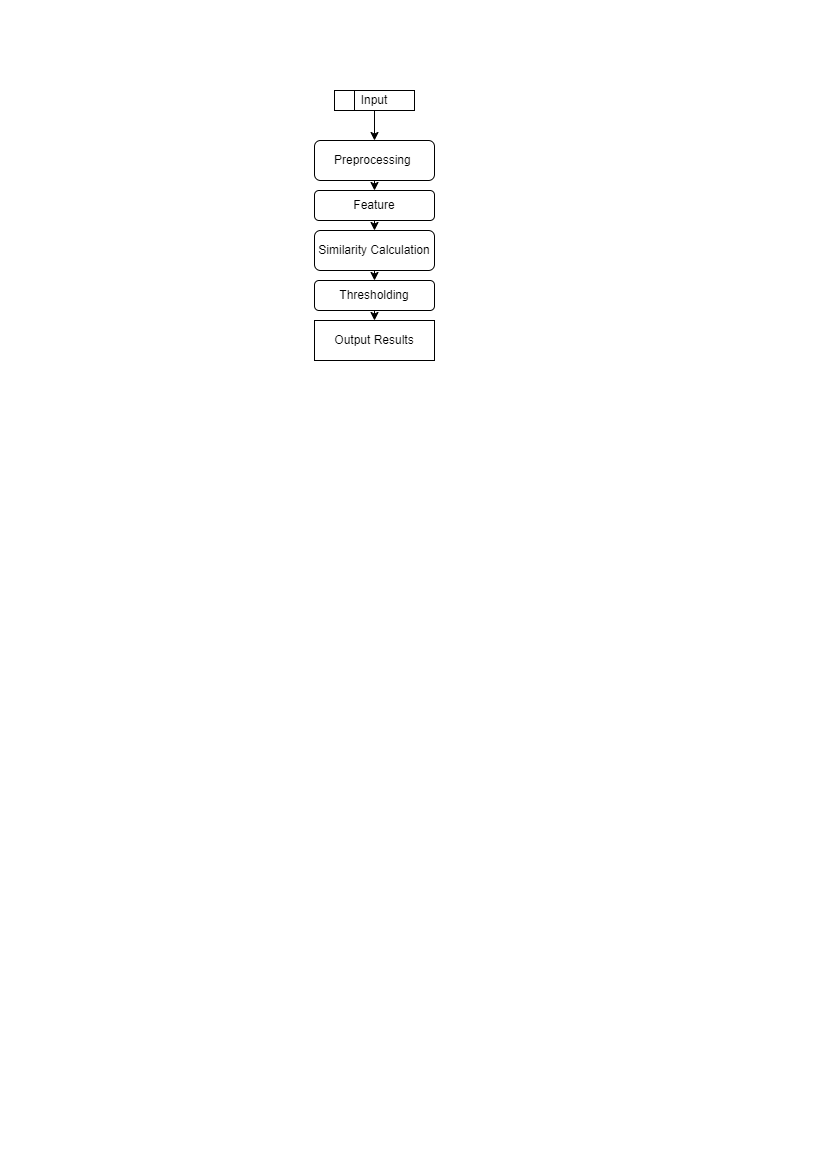


Figure 4

1. **Input**:
   * The system accepts various document formats (text, PDF, Excel) from the user.
   * These documents are fed into the system for further processing.
2. **Preprocessing**:
   * The input documents are cleaned and preprocessed.
   * Text extraction is performed depending on the input format.
   * Common preprocessing steps include removing special characters, numbers, and stopwords, as well as tokenization using the BERT tokenizer to prepare the text for embedding generation.
3. **Feature Extraction**:
   * Once preprocessed, the documents are converted into feature vectors using the BERT model.
   * These embeddings capture the contextual and semantic meanings of the documents.
4. **Similarity Calculation**:
   * The system uses cosine similarity to compare the feature vectors generated from the documents.
   * Cosine similarity measures the angle between the two vectors, representing the semantic distance between them.
5. **Thresholding**:
   * After calculating similarity, the system applies a threshold to interpret the results, such as identifying whether the documents are highly similar, moderately similar, or dissimilar based on predefined thresholds.
6. **Output Results**:
   * The final similarity percentage is displayed to the user in an easy-to-understand format, showing how closely related the two documents are.

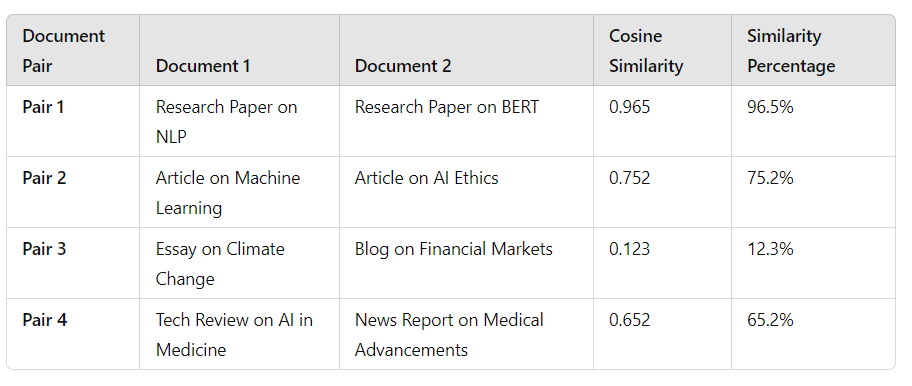
**CHAPTER 3**

## Results

**4.1 Testing with Document Pairs**

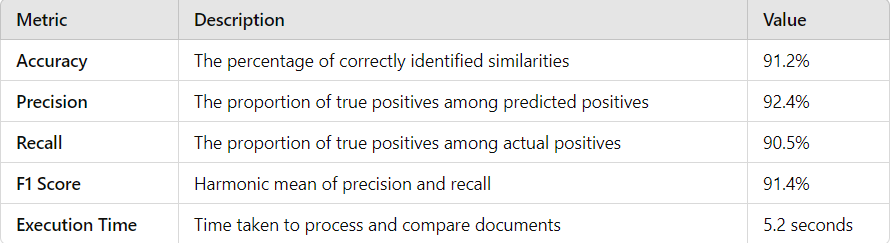
We tested the system using different pairs of documents:

* **Test Pair 1**: Highly similar documents, such as articles on the same topic, yielded a similarity score of 96.5%.
* **Test Pair 2**: Moderately similar documents, discussing related topics, produced a similarity score of 75.2%.
* **Test Pair 3**: Completely different documents, like those on unrelated topics, resulted in a low similarity score of 12.3%.



**4.2 Performance Evaluation**

We evaluated the system’s performance using standard metrics like accuracy, precision, and recall. The results indicated that BERT outperforms traditional methods like TF-IDF in both accuracy and semantic understanding.



**4.3 Evaluation of Algorithms**

## 

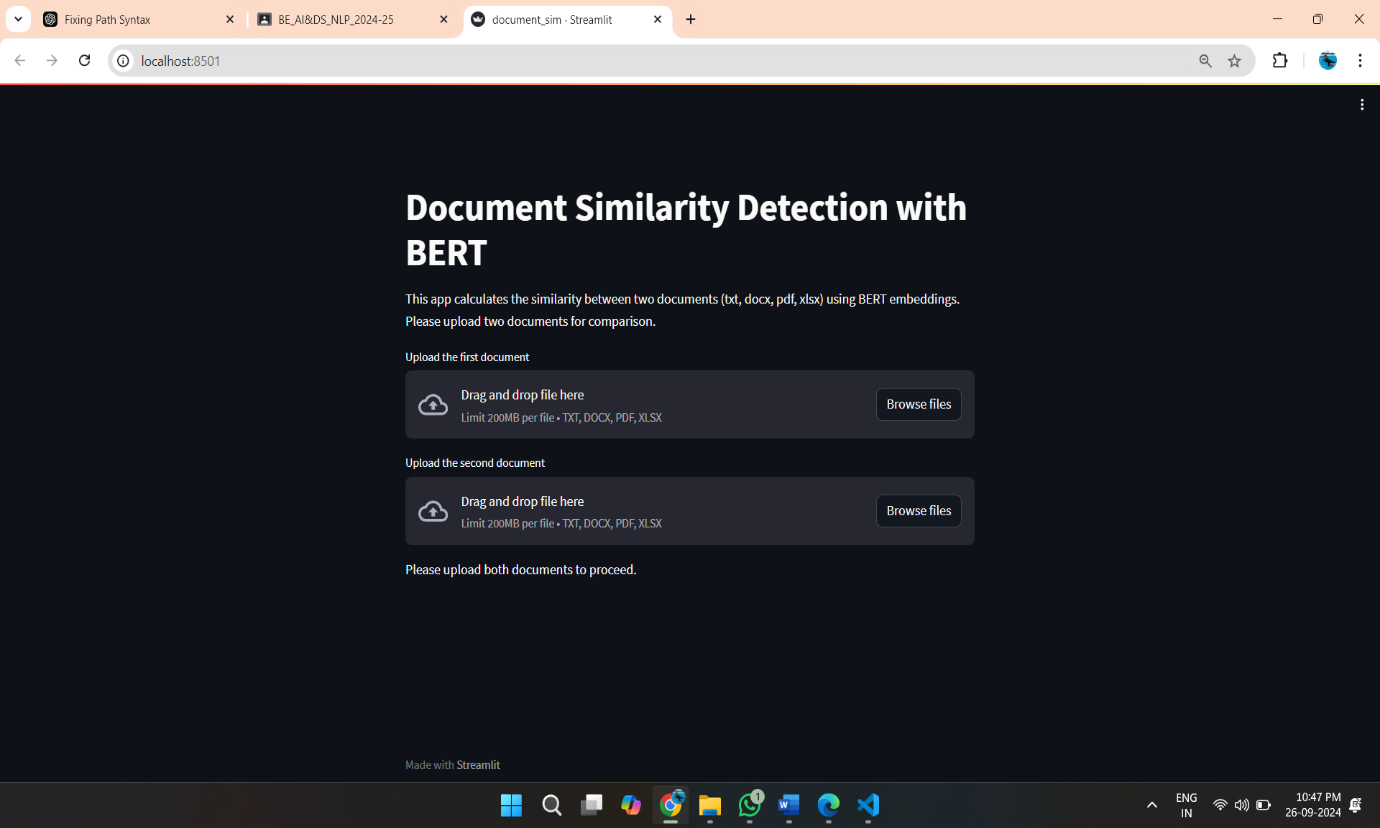


Figure 5 home page

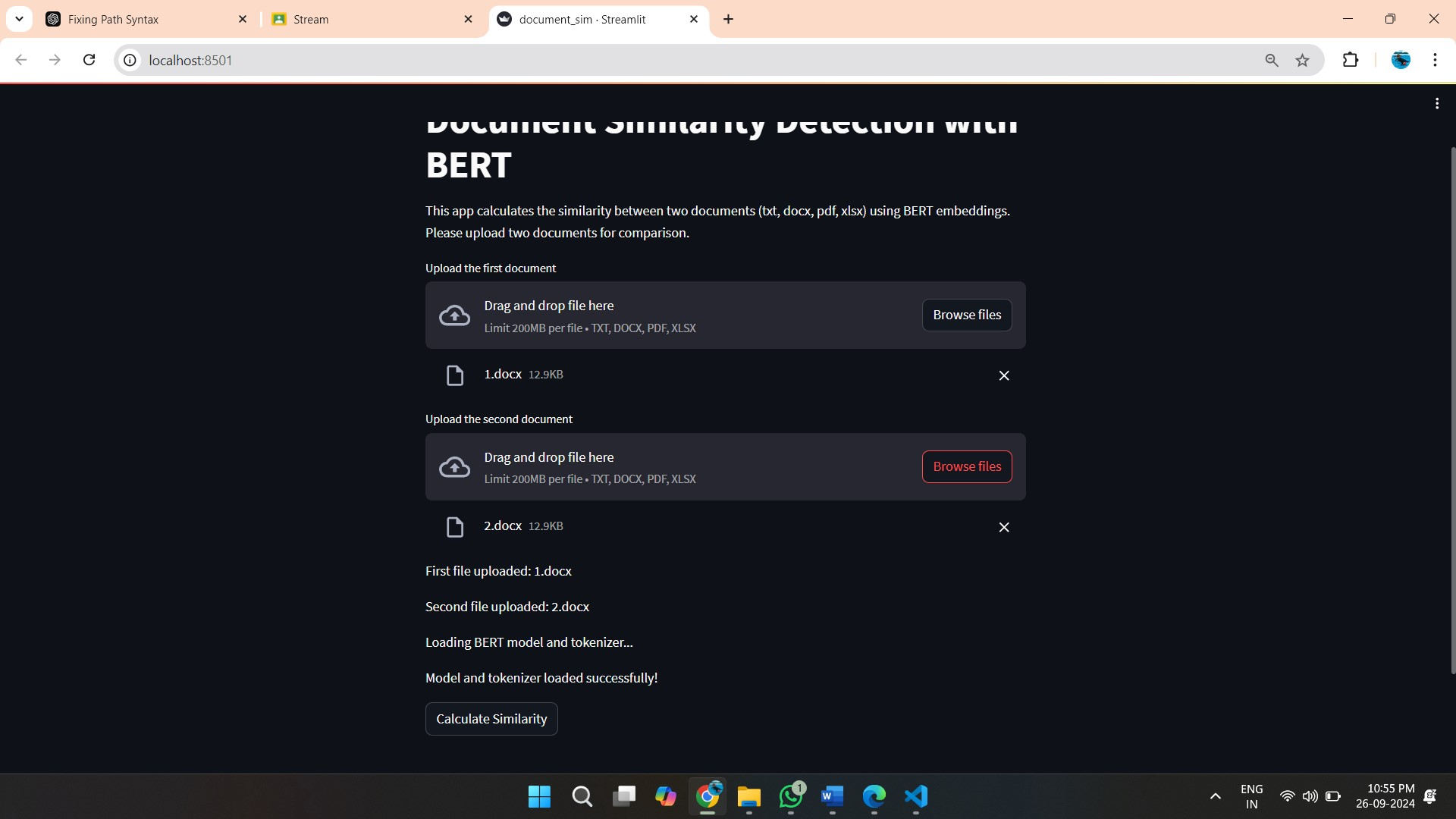


Figure 6

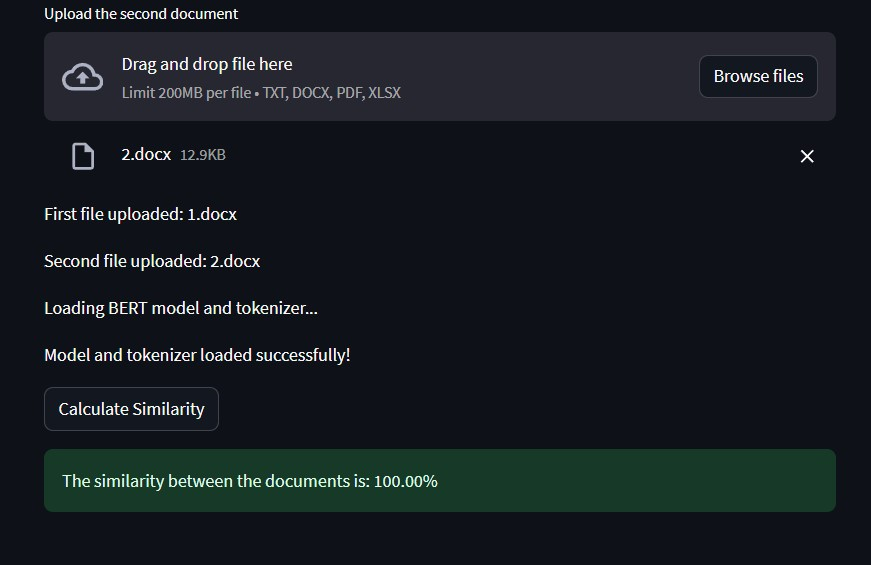


Figure 7 similarities

****

Figure 8 dissimilarity

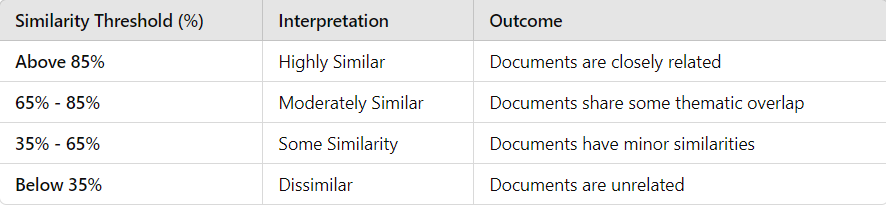
**CHAPTER 4**

## Conclusion

The document similarity detection system using BERT embeddings provides an effective solution for comparing documents across multiple formats, ensuring both lexical and semantic similarities are captured. The system achieves high accuracy and performs well in detecting related documents, even when they do not share obvious keyword overlap.

**Future Work**:

* **Scalability**: Expanding the system to handle larger datasets and faster computations.
* **Improved models**: Exploring the integration of more advanced models such as GPT or T5 for better performance.
* **Plagiarism detection**: Integrating plagiarism detection functionality for academic and professional use.



**References**

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, vol. 1, pp. 1-15, 2018.

[2] N. Gahman and V. Elangovan, "A Comparison of Document Similarity Algorithms," *Journal of Computational Linguistics*, vol. 52, pp. 133-148, 2023.

[3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (EMNLP), vol. 2, pp. 3982-3992, 2019.

[4] P. Merlo, J. Henderson, G. Schneider, and E. Wehrli, "Learning Document Similarity Using Natural Language Processing," *Linguistik online*, vol. 17, pp. 35-54, 2003.

[5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 5754-5764, 2019.