**Document Similarity Checker Using MinHash and Gradio Interface**

**Project Aim**

The aim of this project is to design and develop a reliable and efficient tool for DOCX document similarity calculation. With the use of the MinHash algorithm, the project approximates Jaccard similarity, which is suitable for plagiarism detection, document version comparison, and text clustering. Simplicity and accessibility guide the construction of the project with the feature of an interactive Gradio web interface to upload files and visualize similarities.

The project addresses this need for scalable and accurate comparison of texts in academic, professional, and personal contexts by supplying users with an automated text similarity checker.

**Methodology**

The tool is implemented using the following key steps:

1. **Text Extraction**:  
   The Python python-docx library extracts textual content from the DOCX files uploaded by the user. This ensures seamless document format handling, enabling accurate paragraph-level text extraction.
2. **Shingling**:  
   After text extraction, the document content is divided into fixed-length character sequences called shingles. This implementation sets the shingle size (k) to 5. Shingling captures the contextual and structural details of the document, serving as input for the MinHash process.

Example for text "hello world" with k=5:

* + Shingles: "hello", "ello ", "llo w", "lo wo", "o wor", " worl", "world".

1. **MinHash Algorithm**:  
   The MinHash algorithm generates hash-based signatures for the shingles of each document. By applying 128 hash permutations, the algorithm creates compact representations of the original text, enabling efficient computation of the Jaccard similarity.
   * **Jaccard Similarity**: The MinHash signatures are used to calculate the similarity between the documents.

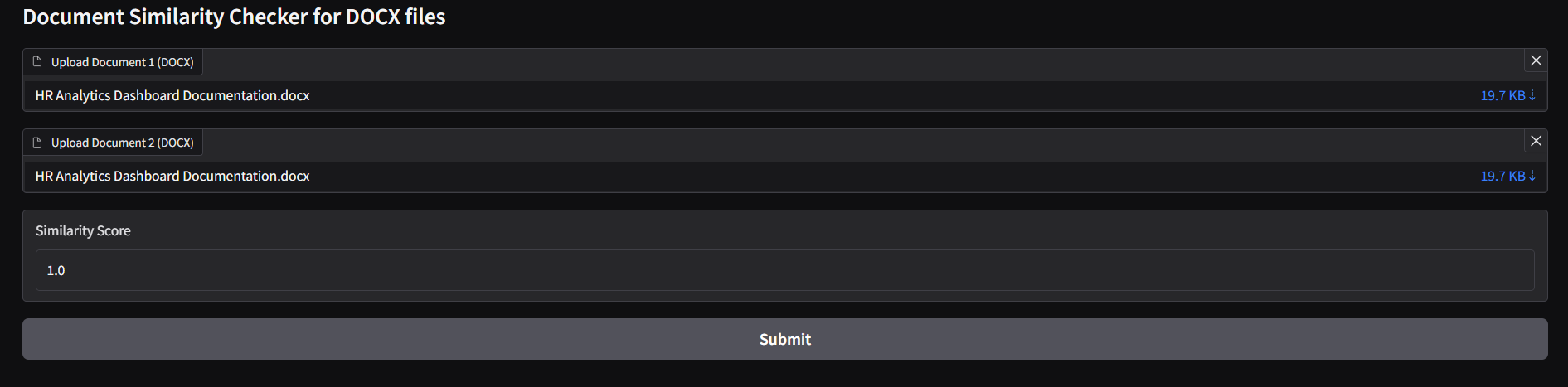
Jaccard similarity is defined as: J (A, B) = ∣A∩B∣/∣A∪B∣

**Gradio Interface**:  
A web-based interface is built using Gradio to allow users to upload two DOCX files and view the calculated similarity score in real-time.

* + **Inputs**: Two DOCX files.
  + **Output**: A similarity score ranging from 0 (completely different) to 1 (identical).

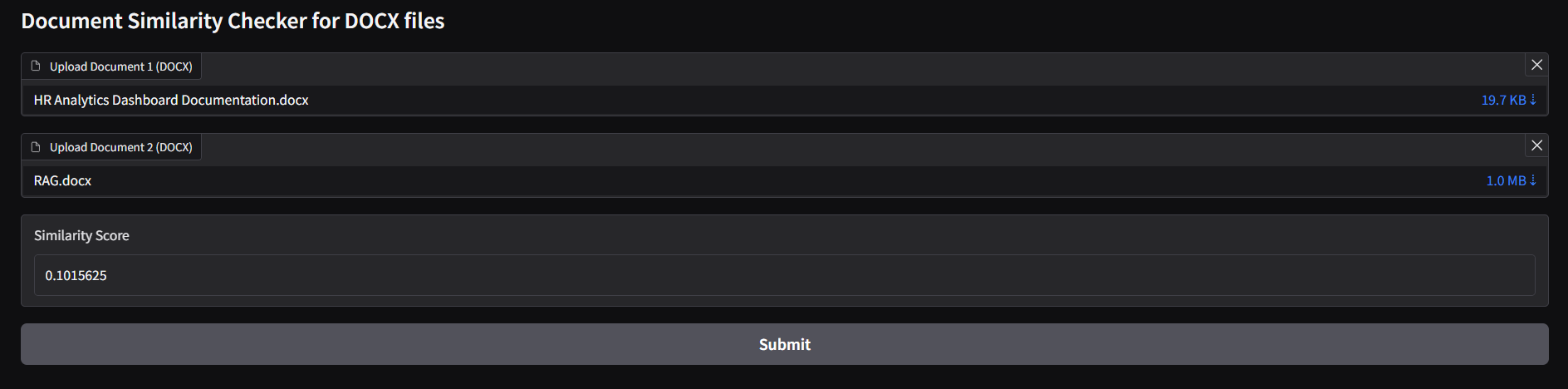
**Results and Observations**

1. **High Similarity for Identical or Highly Similar Documents**:  
   The tool produces similarity scores close to 1 for documents with overlapping text. For example, two versions of the same document with minor edits yield a similarity score of 0.95 or higher.

**Screenshot Example**:

1. **Low Similarity for Dissimilar Documents**:  
   When comparing unrelated documents, the tool generates a similarity score near 0. This demonstrates its effectiveness in distinguishing different textual content.

**Screenshot Example**:



**Applications**

This project is applied in various domains.

* **Plagiarism Detection:** Detects reuse or copying of texts in academic or professional documents.
* **Document Clustering:** Group similar documents into categories for research or content.
* **Version Comparison:** Comparing draft or different versions of a paper to follow changes over time.
* **Content Verification:** Authenticating the originality of text from various sources.

**Technical Highlights**

* **Efficiency:** MinHash algorithm reduces the computational complexity while preserving accuracy for large document sizes.
* **User Friendly Interface:** Gradio supports drag-drop functionality that comes as intuitive for the minimum of user expertise.
* **Scalability:** The modular design allows future improvements, such as support for additional file formats or the integration of Locality-Sensitive Hashing (LSH) for large-scale comparisons.

**Project Links**

* **GitHub Repository**:  
  [Document Similarity Detection Using Shingling and Locality-Sensitive Hashing](https://github.com/srikta/Document-Similarity-Detection-Using-Shingling-and-Locality-Sensitive-Hashing.git)
* **Hugging Face Space**:  
  [Document Similarity Checker](https://huggingface.co/spaces/Sourikta/doc_similarity)

**Conclusion**

This project shows a practical and feasible solution for document similarity detection using MinHash. The tool is very versatile in its applications, which range from academics to industries. The Gradio interface makes it accessible for non-technical users to use the similarity analysis.

Future Developments

* Supporting multiple file formats (e.g., PDF, TXT).
* Using Locality-Sensitive Hashing (LSH) for scalability.
* Providing visualization features to see the differences between documents.

This tool's success demonstrates the potential for combining efficient algorithms like MinHash with user-friendly platforms like Gradio for real-world applications.