Automatic Key Detection Across Structured and Unstructured Text

Mawil Hasan, Aditya Parameswaran, Alvin Cheung, Yiming Lin

Background

PDF's are complex.

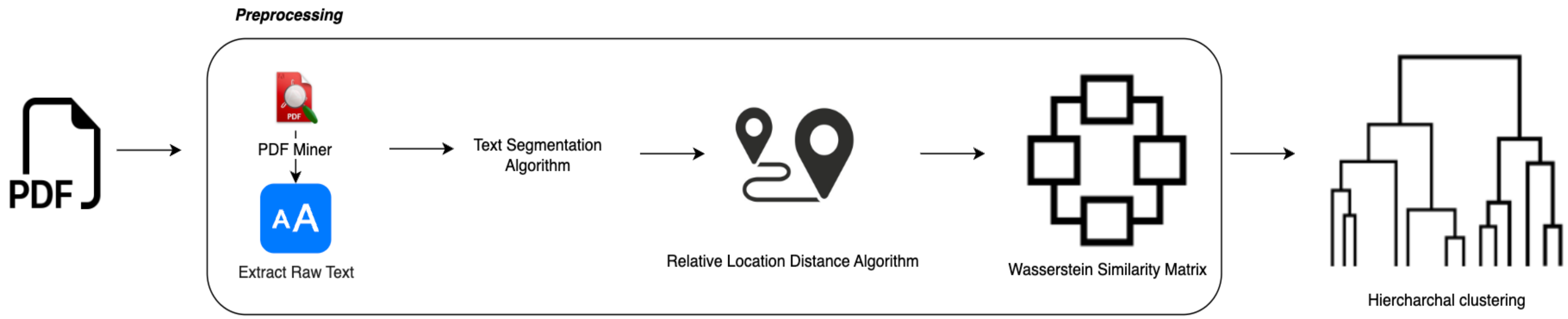
- Data can be formatted in an unstructured or structured format
- PDF's can be image-based or non-image based
- Contain complex visual cues

It's easy to suggest using a large language model (LLM) to process a PDF for table extraction – a user friendly way to understand a document.

However, even LLM's have a hard time formulating a tabular structure for user's readability due to unstructured or structured data formulated in a PDF.

Goal: Devise a cheap and efficient algorithm that formulates a tabular structure easier for users to understand PDF documents

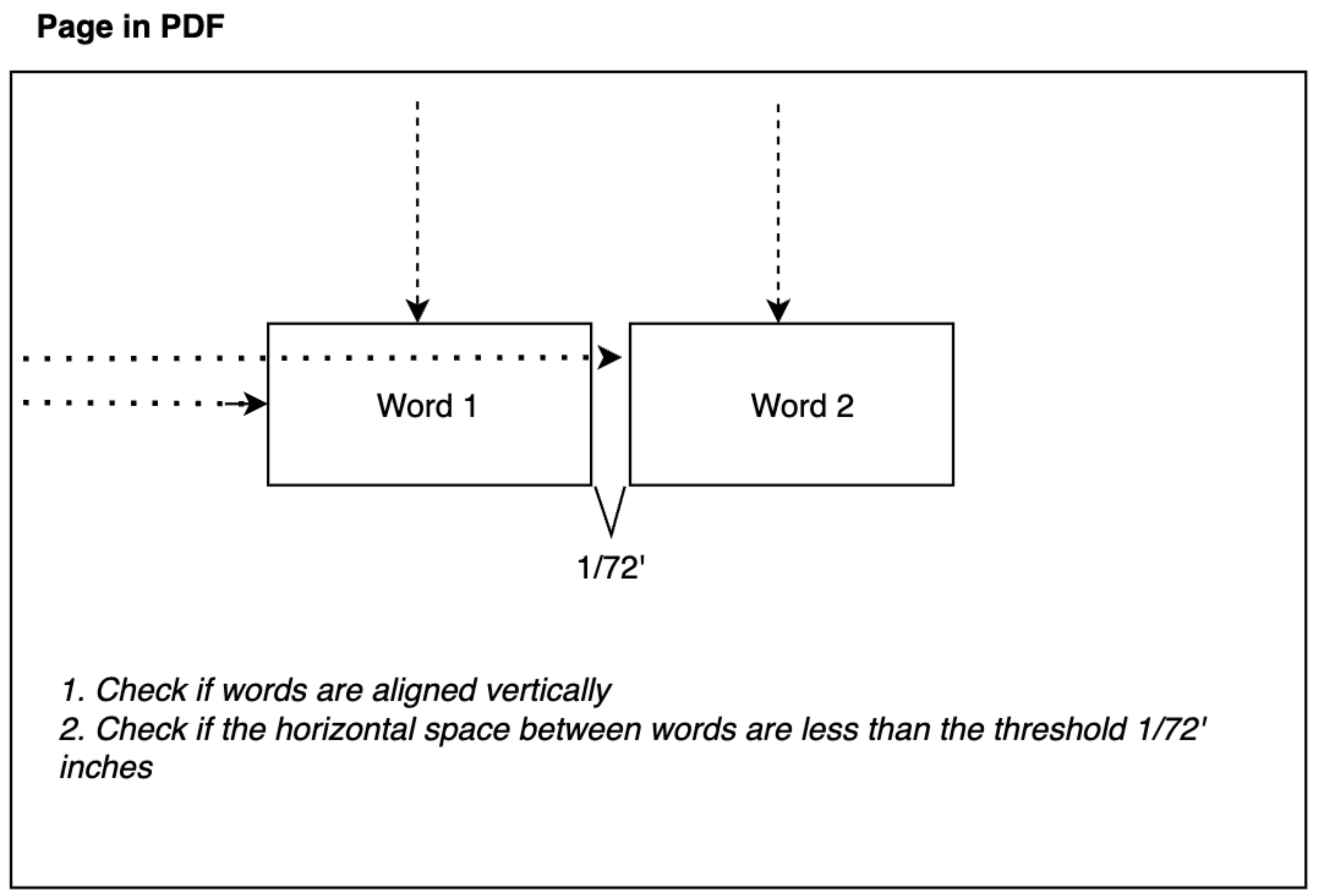
Workflow



Preprocessing
PDF
PDF Miner
Extract Raw Text
Text Segmentation Algorithm
Relative Location Distance Algorithm
Wasserstein Similarity Matrix
Hiercharchal clustering

Preprocessing Algorithms

Text Segmentation



1. Check if words are aligned vertically
2. Check if the horizontal space between words are less than the threshold 1/72' inches

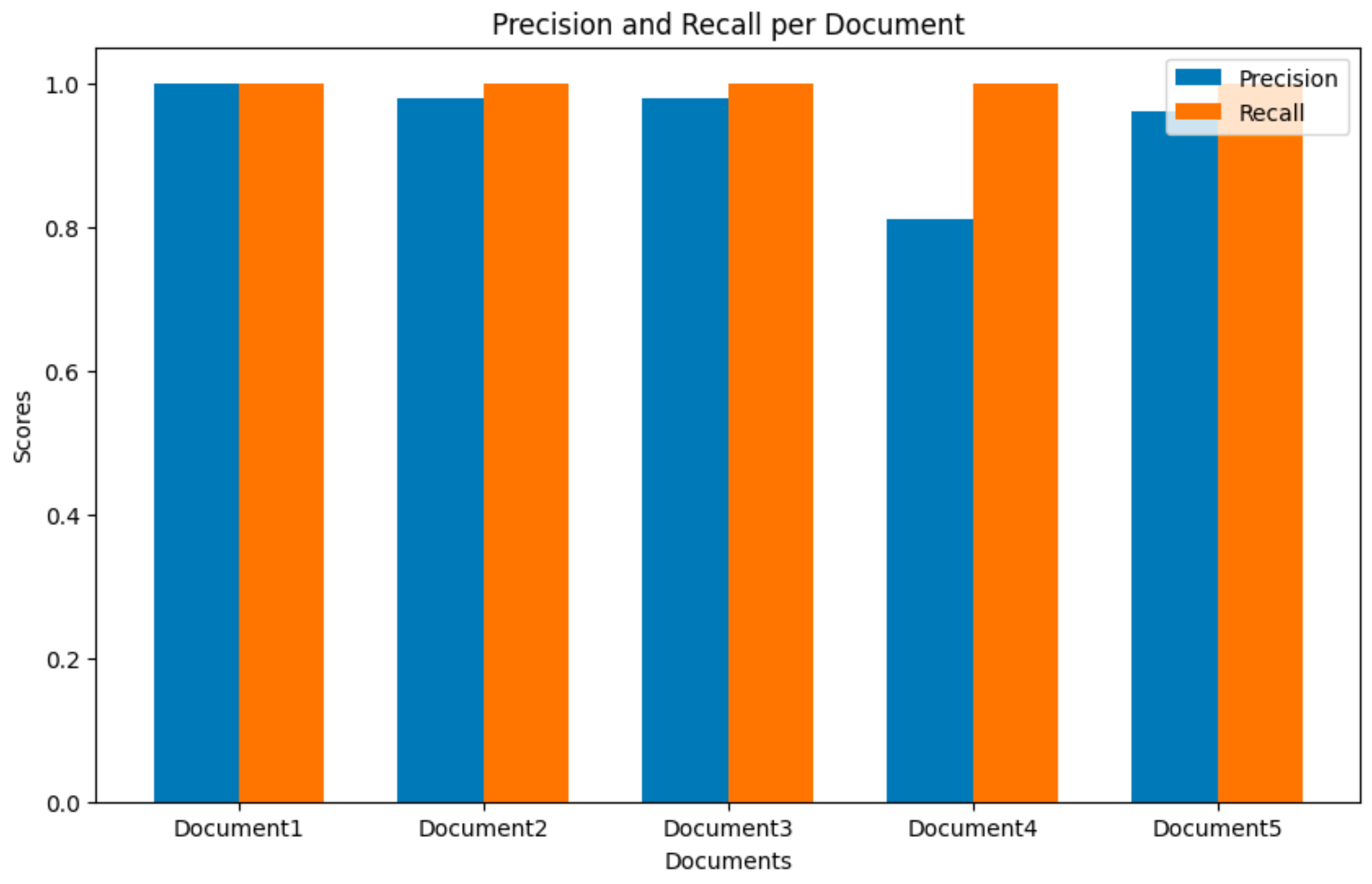
Relative Location Algorithm

Create a list that maps each phrase in the document to its occurrence line numbers.

Wasserstein Similarity Matrix

Using the vectors, compare the Wasserstein per list

Results



Findings:

When predicting the keys in these PDF documents, we notice the following edge cases:

1. Some keys can be values
2. Some phrases repeat relatively the same location as keys as many times
3. Metadata (such as headers and footers) repeat as many as keys

Future Works

- Extend this phase to key-value pair detection
- Support image-based documents (extending the preprocessing pipeline

Github Repo Link:

https://github.com/ucbepic/pdfReverseEngineer