IT-350 Assignment 2

Abhayjit Singh Gulati

February 7, 2024

Abstract

This is the report for Assignment-2 of the IT-350 course. The code for the assignment can be found on the following: [GitHub]

1 Problem Statement

To analyze the similarity among multiple documents of a News Corpus using the min-hash algorithm. Utilizing Jaccard Similarity and Dice Similarity for the analysis, plotting the results and drawing inferences.

2 Dataset

The dataset chosen for this problem is the AG News Classification Dataset, which is available on Kaggle. The dataset can be accessed using the following link: [Dataset].

The dataset consists of 4 classes and each class consists of 30,000 news items. The dataset is visualised in Figure 1.

As we only need three categories for this assignment, we exclude news items of category 3. Additionally, the number of data items are reduced to 100 per category, as specified by the assignment. The visualisation is in Figure 2.

The number of characters per document is visualised in Figure 3.

3 Background

3.1 Shingling

Shingling is the most effective way to represent documents as sets, for the purpose of identifying lexically similar documents. This is done by constructing from the document the set of short strings that appear within it. On doing so, the documents that share pieces as short as sentences or even phrases will have many common elements in their sets, even if those sentences appear in different orders in the two documents. This assignment utilises section the most common approach, k-shingling.

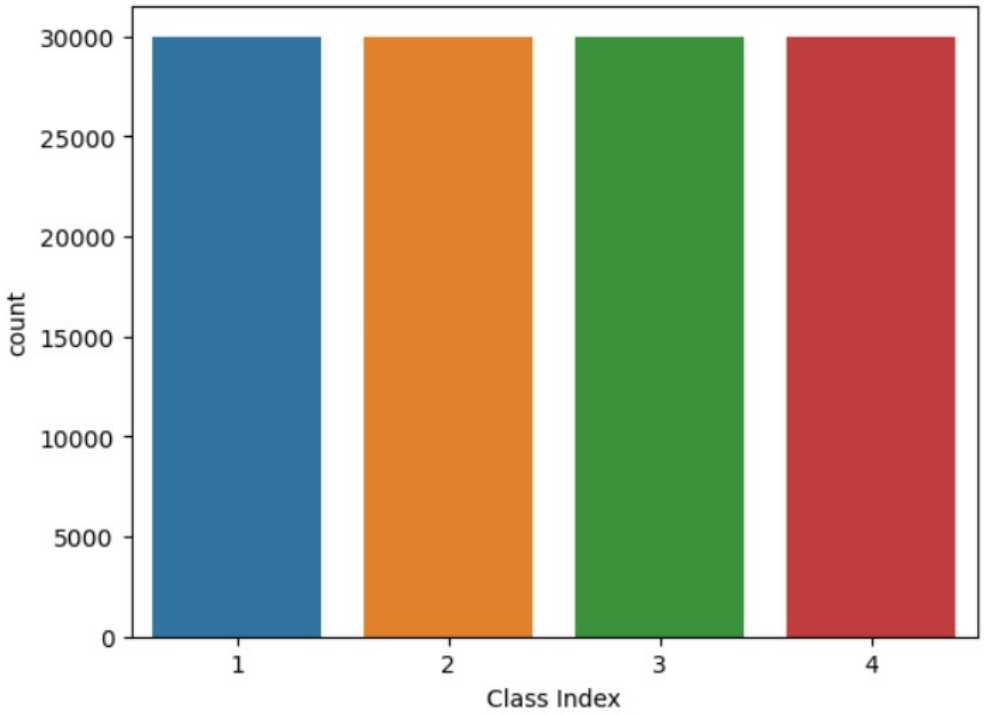

Figure 1: Dataset Visualisation

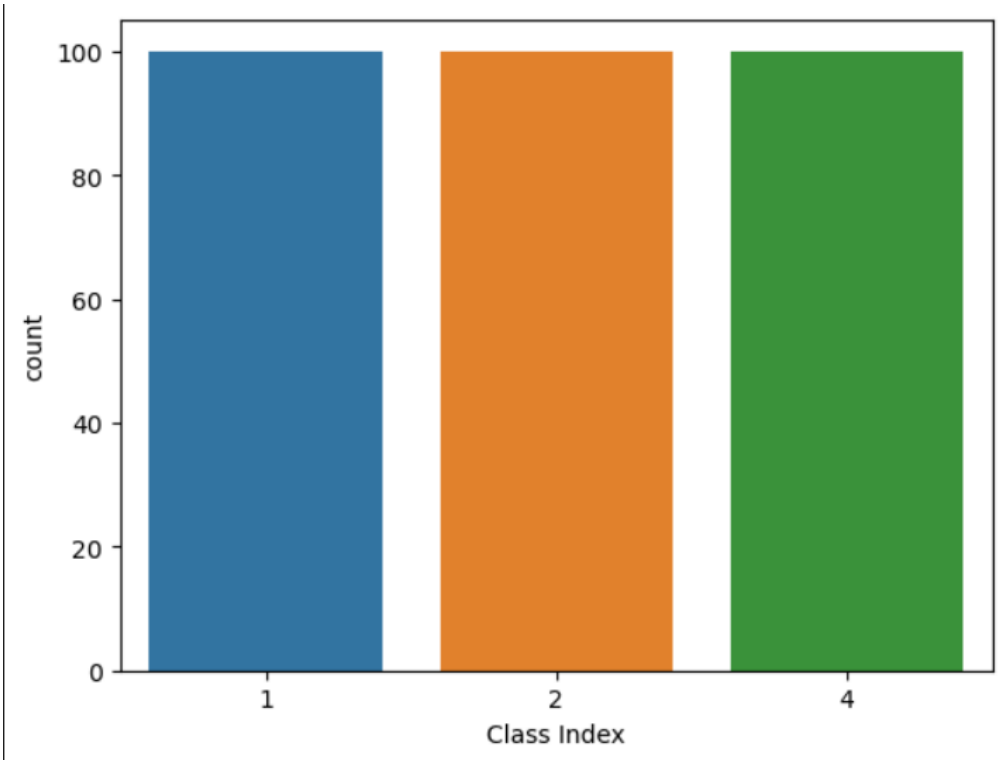

Figure 2: Dataset Visualisation after reducing the number of documents per class to 100.

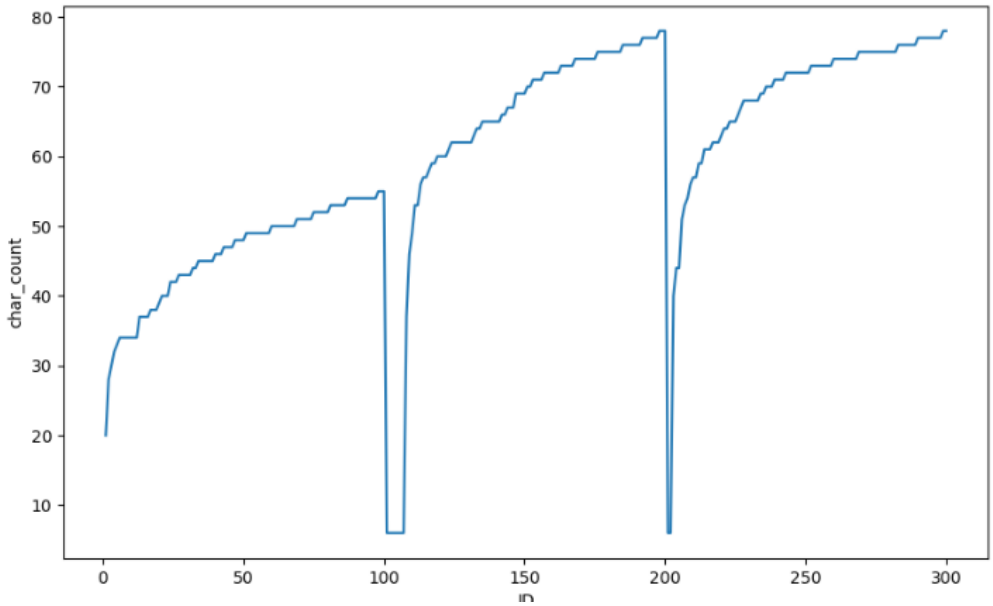

Figure 3: Character Count Visualisation

3.2 Min-Hashing

This assignment utilises the algorithm provided in section 3.3.5, Computing Minhash Signatures in Practice, of the textbook `Mining of Massive Datasets`.

3.3 Jaccard Similarity

The Jaccard similarity measures the similarity between two sets of data to see which members are shared and distinct.
The Jaccard Similarity for column column matrix is calculated as follows:

1. The intersection is the number of rows where both columns have a value of 1, indicating that the corresponding k-shingle appears in both documents.

2. The union is the number of rows where either column has a value of 1, indicating that the corresponding k-shingle appears in at least one document.

3. The algorithm then returns the intersection divided by the union, which is the Jaccard similarity

4. If the union is zero, the algorithm returns zero to avoid division by zero.

The Jaccard Similarity for signature matrices is calculated as follows:

1. The algorithm calculates the Jaccard similarity by dividing the number of rows where the two columns have the same minhash value by the total number of rows in the signature matrix.

3.4 Dice Similarity

The Dice similarity, also known as the Sørensen–Dice index or simply Dice coefficient, is a statistical tool which measures the similarity between two sets of data.
The equation for the Dice Similarity is as follows:

$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|}$$

where,

1. X and Y are two sets
2. a set with vertical bars either side refers to the cardinality of the set, i.e. the number of elements in that set, e.g. —X— means the number of elements in set X
3. + is used to represent the intersection of two sets, and means the elements that are common to both sets

4 Procedure

1. Create character shingles for each document.
2. Character shingles are generated with the values of $k = 5, 8$, and 10.
3. Perform Min-Hashing using the inclusion probability method.
4. Build the signature matrix with retention values set at $10\%, 20\%$, and 30% of the number of shingles.
5. Examine Jaccard Similarity and Dice Similarity for both the column-column matrix and the signature matrix.

5 Inferences

From the plots below, the following inferences can be drawn:

1. The shapes of the plots for the column-column matrix and signature matrices are similar. This shows that we can effectively capture similarity using Signature Matrices.
2. The captured similarity values are inversely proportional to the value of k. In other words, the similarity values are highest for k=5 and decrease as we increase the value of k.
3. Increasing the percentage of min-hashing results in a slight increase in the captured similarity values.
4. Items of Category 1 are similar to each other while items of Category 3 are similar to each other.

6 Further Work

After understanding the min-hashing technique using character shingles, another method of shingling, the word shingling was used for analysis. It was found that in the case of word shingling, the similarity captured reduces drastically and it is not effective in analysing the similarity between documents. The code and plots for the same can be accessed on the Github Repository

7 Plots

Following are the set of plots obtained using multiple similarity metrics.

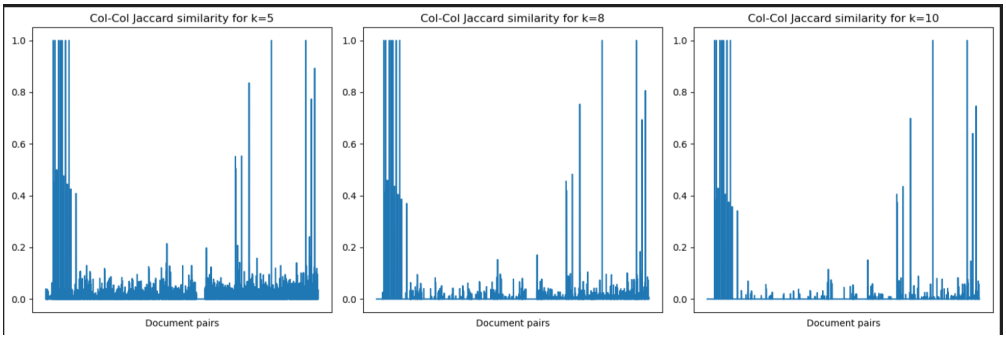

Figure 4: Column Column Similarity

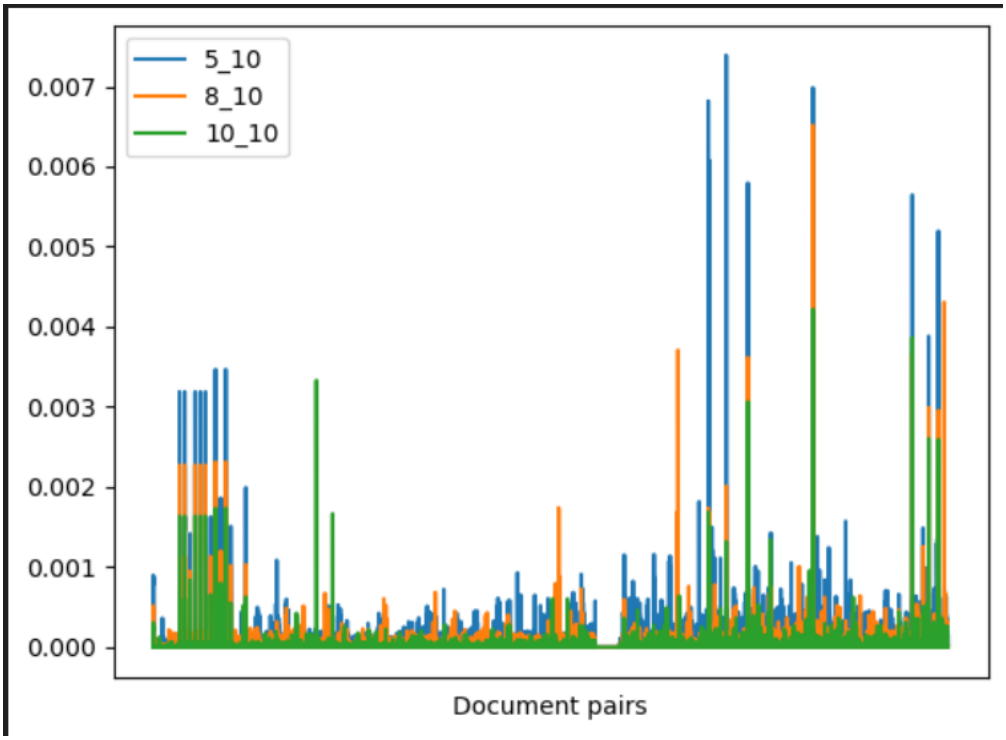

Figure 5: Jaccard Similarity for Signature Matrix

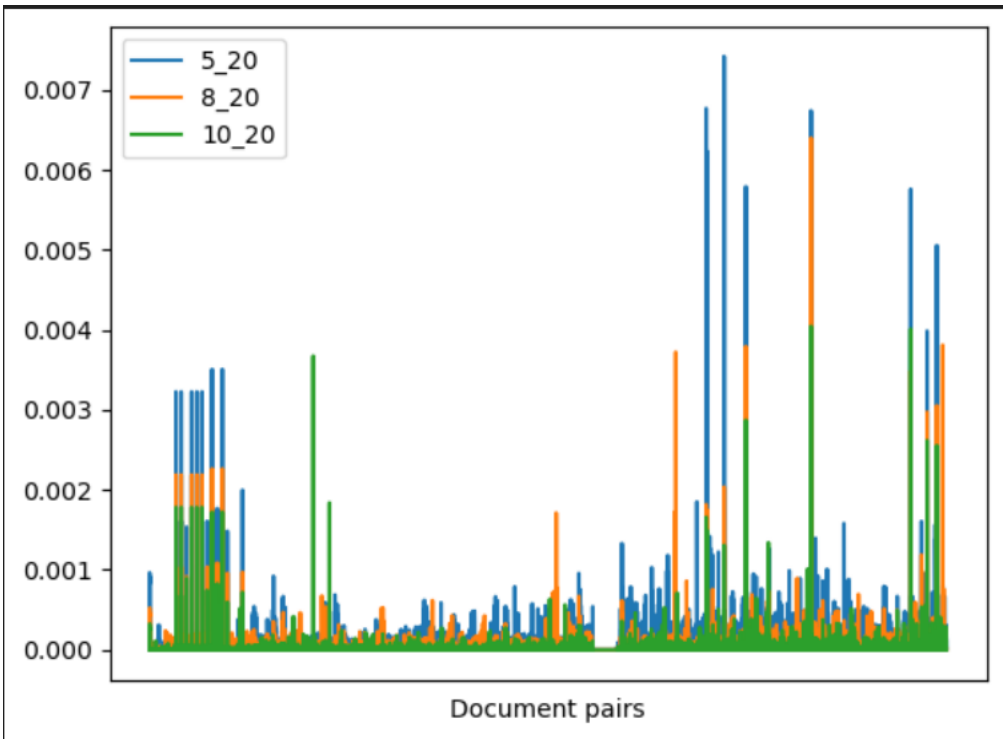

Figure 6: Jaccard Similarity for Signature Matrix

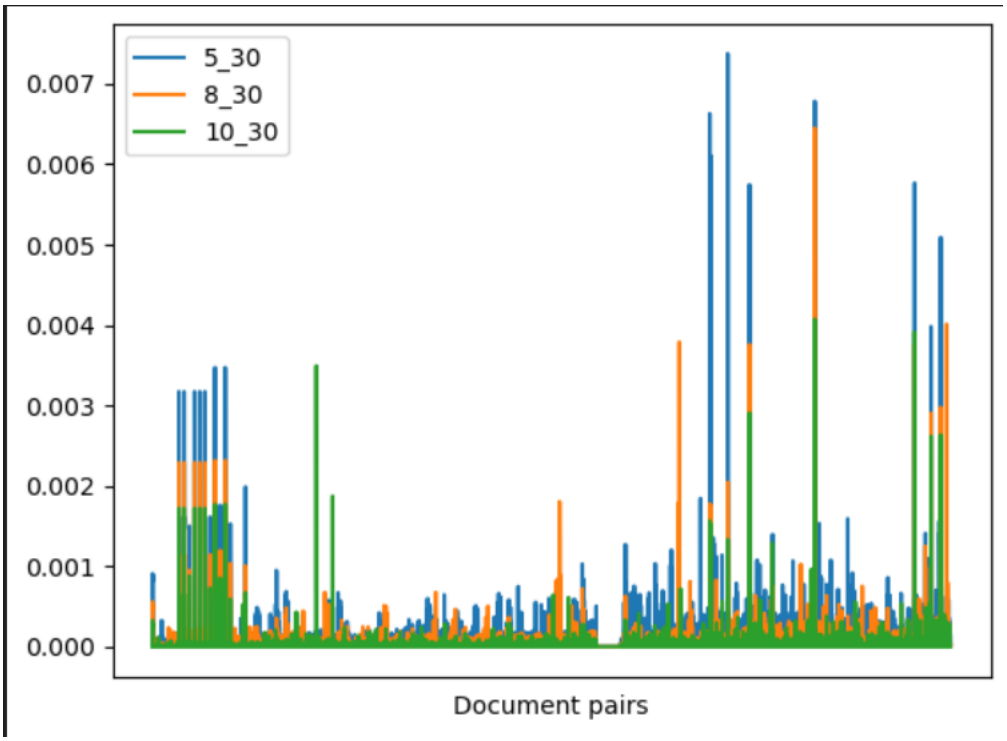

Figure 7: Jaccard Similarity for Signature Matrix

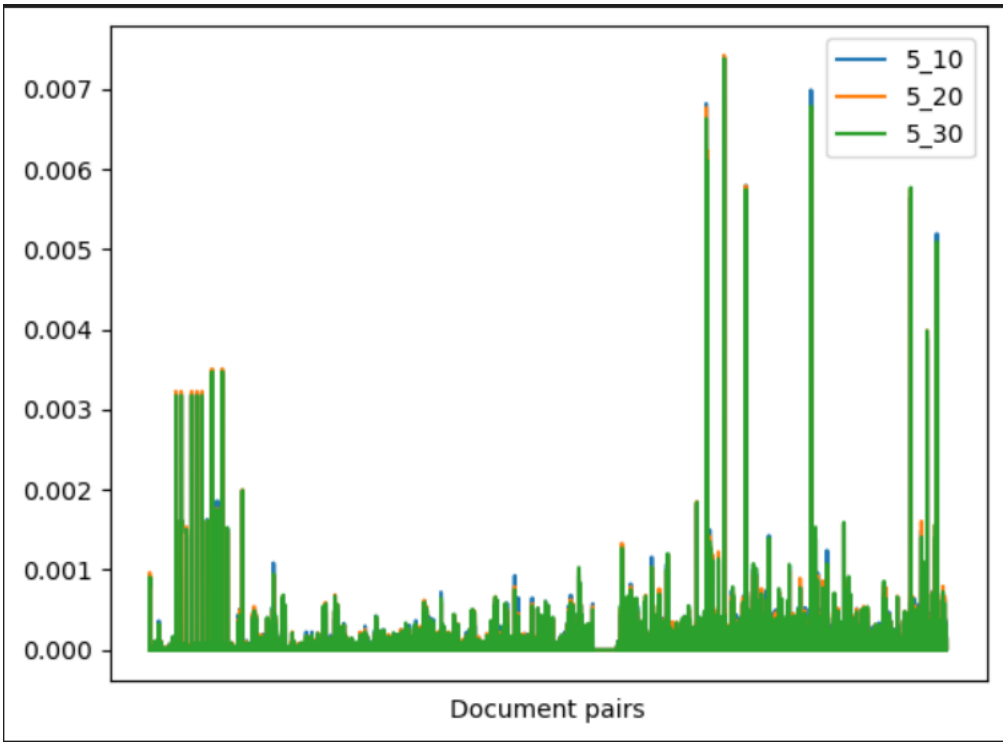

Figure 8: Jaccard Similarity for Signature Matrix

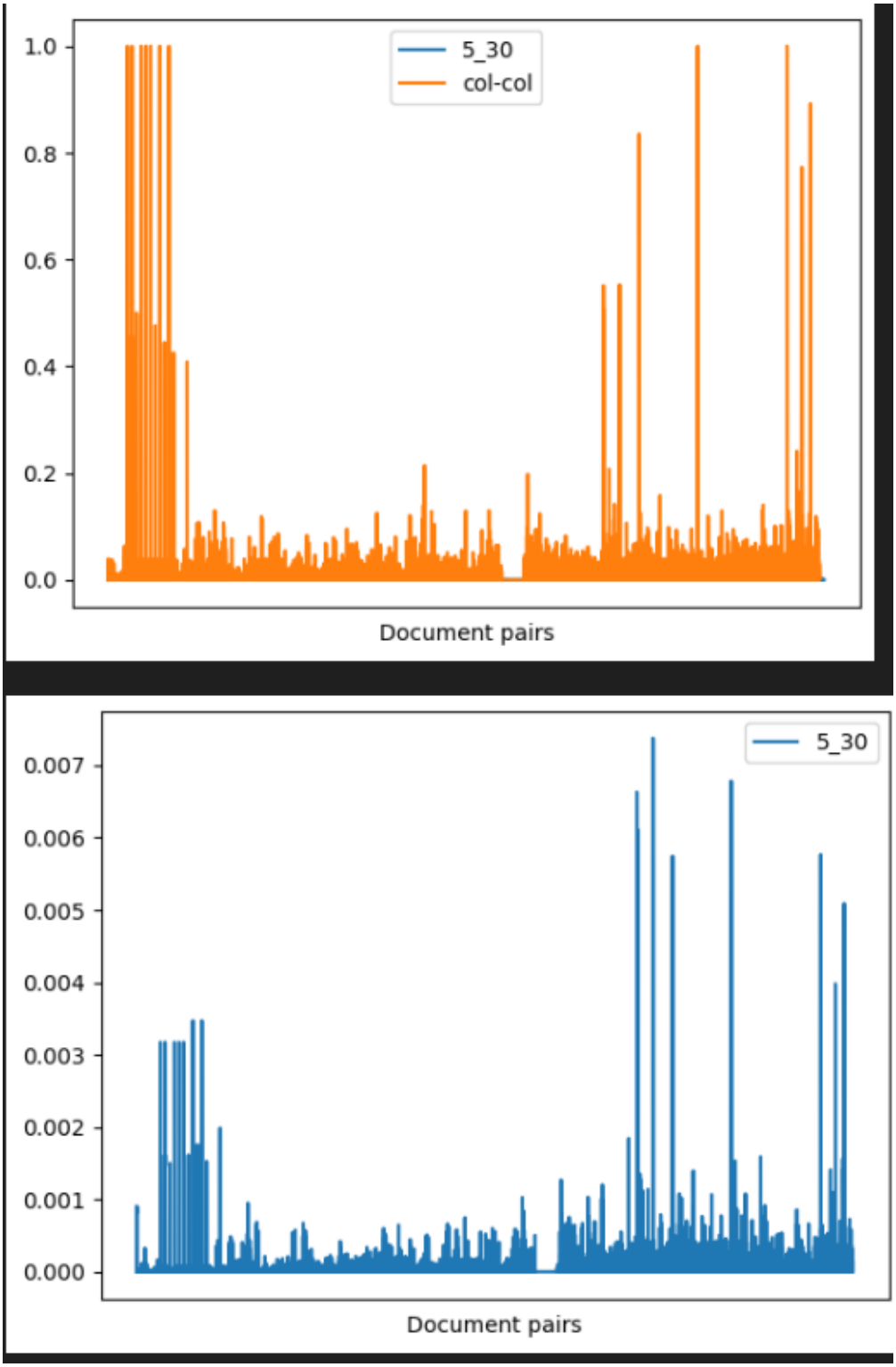

Figure 9: Capturing Jaccard Similarity

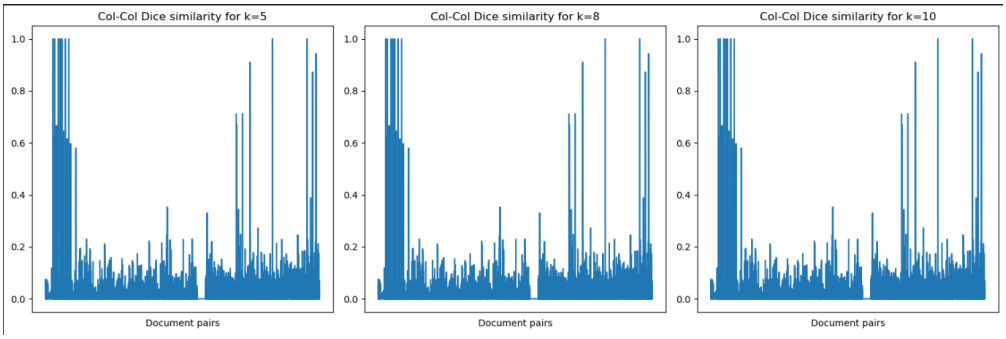

Figure 10: Dice Similarity for Column Column Matrix

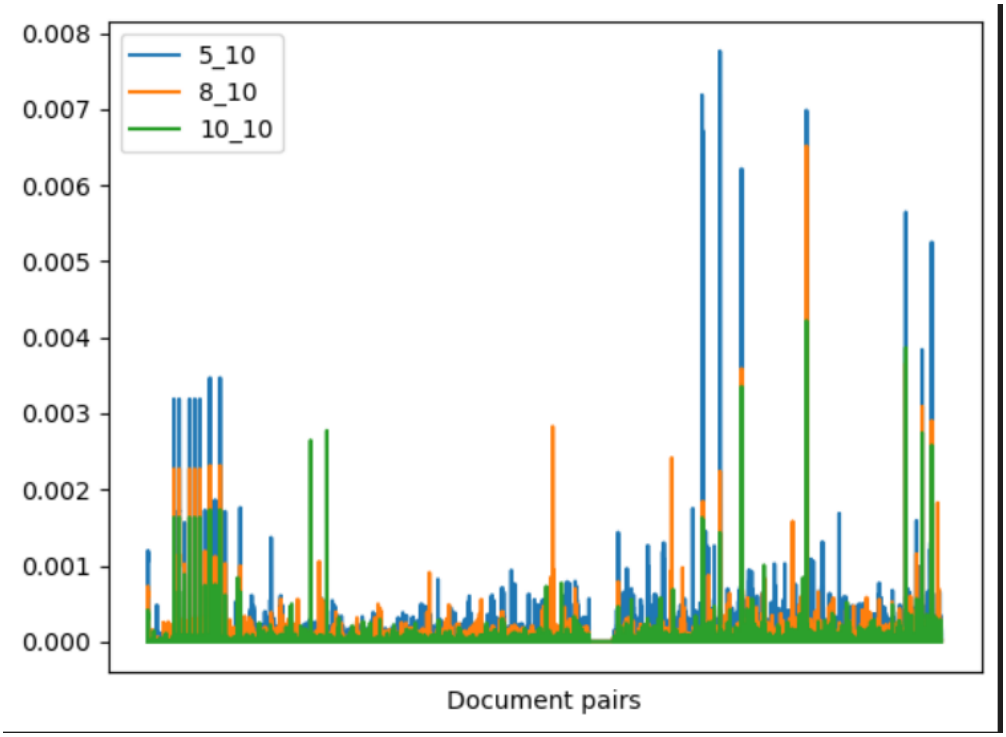

Figure 11: Dice Similarity for Signature Matrix

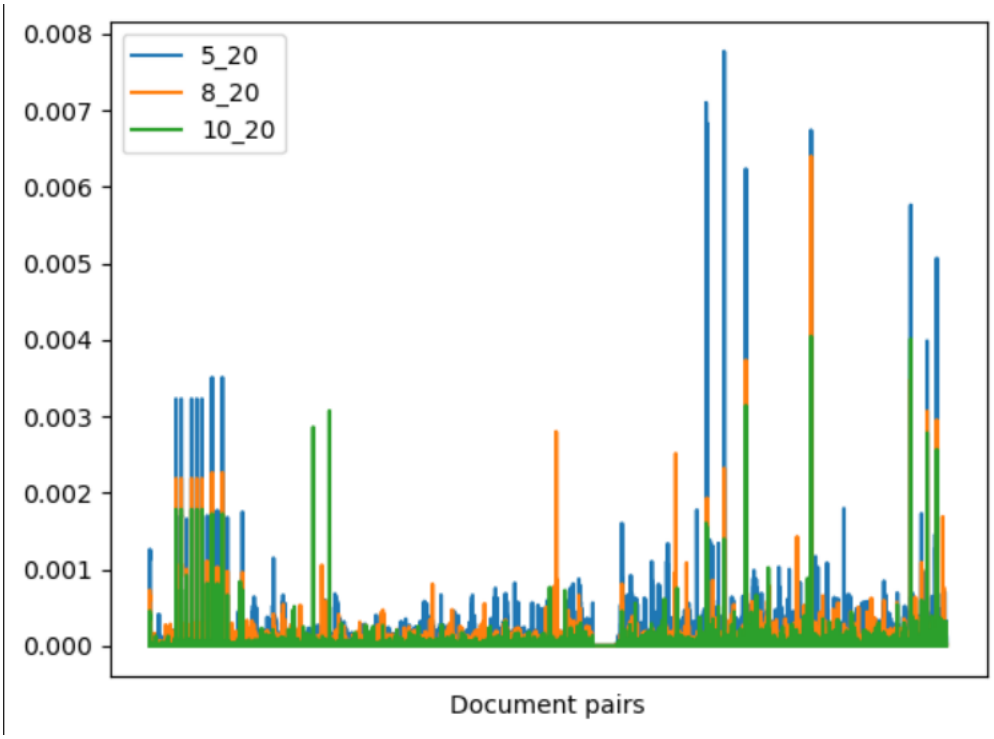

Figure 12: Dice Similarity for Signature Matrix

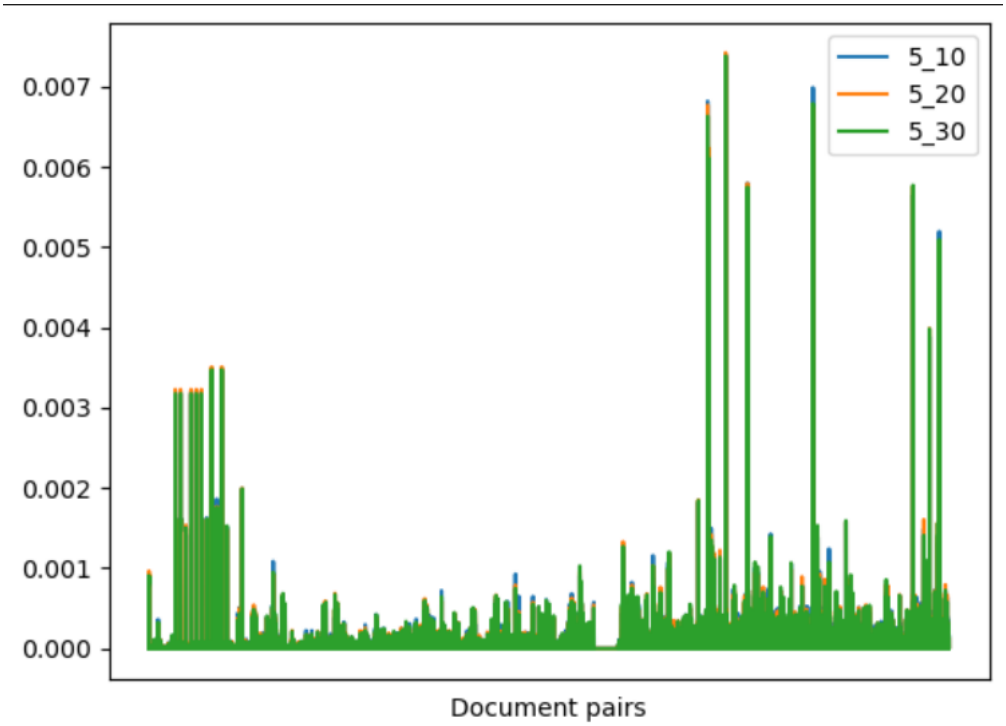

Figure 13: Dice Similarity for Signature Matrix