Duplicate Discussion Detection

You have been provided a text file (*discussions.thorn*) that is a thorn separated version of some discussion comments from the online section of this course. Assume the first comment has an ID of 1. Your task is to exercise your GNU/Linux knowledge and create a script to perform a pairwise comparison of all the discussion comments. The goal is to produce a ranked list of pairwise discussion pairs with a similarity score. The first pair of discussion comments should be the most similar and the last pair, the least similar.

Your similarity score should support the following property:

```
Sim(bb,bb) > Sim(ab,bb) > Sim(aa,bb);
```

For any arbitrary a and b. Meaning Identical strings should have a higher similarity when compared to two strings that are completely different.

To compute similarity, use the GNU/*Linux diff* command. As you know, this command lets you compare two files for textual similarity. Review the capability of this tool and define your own approach for comparing similar documents. You can use word, space, line or other comparisons types. You can break ties arbitrarily, be careful with Unicode characters across comparisons.

A helpful command is the one below.

```
diff -u <(echo "aba"| fold -w1) <(echo "abaa" | fold -w1)
```

This command uses three GNU/Linux Commands, namely, diff, echo, and fold. This command compares two strings, “aba” and “abaa” character by character. Using this command, (run multiple times, with and without the “-u” flag) you can create a distance metric similar to the Jaccard measure.

Feel free to use Python (os.system) to run the GNU/Linux commands needed to create the discussion comment pairs. You can also complete the assignment using one bash/sh file.

Print the discussion comment pairs to stdout. The output for should be one pair id per line separated by a space. You may optionally include the value of your comparison after each id pair. Below is an example output.

```
2 4 0.523
4 1 0.45
…
```

Describe your approach and assumptions in your README file. Give clear directions on how to replicate the processing of the *discussions.thorn* file. Feel free to use the discussion board to bounce ideas off classmates.

Your submission should be a gzipped file (.gz) containing your README, the thorn file, and your script (either .py or .bash).

Grading criteria

Results are reproducible: 40%
Well-Documented: 40%
Code works correctly: 20%