Data Mining Assignment 1

Name: Rigved Satish Patki
Email Id: patki@kth.se
Group: Assignment-group 8

Topic: Finding Similar Items: Textually Similar Documents

Solution:
For this assignment I am using a dataset create by Paul Clough and Mark Stevenson in University of Sheffield. The reason behind using this data set was that it was specifically collected in order to be used for plagiarism detection. The details of this dataset can be found here .

The code that I have written is in typescript which transpiles to ES6 version of Javascript. The reason behind using typescript is that it type checks javascript reducing the runtime errors and also because I have been using typescript in my work for the past year and I have gotten comfortable with it.

The basic flow of the code is as follows:

- The code start in index.ts file as it is the entry point mentioned in package.json. First we collect all the files from the data folder and convert them to string.
- Next we create a JSON object for each data file. This JSON object consists of the following fields:

```
{
 filePath: string; // absolute path to the data file
 content: string; // content of the data file
 shinglesArray: string[]; // Shingles in form of strings
 shingles: Set<string>; // shingles in the form of set
 hashedShingles: number[]; // hashed shingle using crc32
 minHashSignatures: number[]; // minHash signature of length 128
 lshHashBands: string[]; // lsh hash bands
}
```

- A function is written for each stages of conversions:
    - shingling() function takes in the content from each data files and returns hashedShingles, shinglesArray, shingles
    - minhash() function takes in hashedShingles and converts it into minHash signatures.
    - getHashBands() converts the minHash signatures into hashbands.
- After that we create an JSON object called index consisting of all the hashBands and the files that consisting of the hashBands.

- Then we run a loop on all the documents and compare the Shingles then compare the minHash Signatures and next the LSH hashBands.
- The results are then written down into an output.txt file, sample output file can be found here.
- The config.ts file consists of all the constant values like the value of K, threshold for Jaccard etc, sample config file looks like :

```
export default {
 K: 4,
 JACCARD_SIMILARITY_THRESHOLD: 0.8,
 MINHASH_SIMILARITY_THRESHOLD: 0.8,
 LSH_BAND_SIZE: 4,
 MINHASH_MAX_POSSIBLE_SHINGLES: Math.pow(2, 32) - 1,
 MINHASH_NEXT_PRIME: 4294967311,
 MINHASH_NUMBER_OF_HASHES: 128
};
```

- MakeFile:

```
.PHONY: clean build deploy result install

clean:
  rm -rf build && \
  rm -rf node_modules

install:
  npm install

build:
  npm run build

deploy:
  npm start

result:
  cat output.txt && open output.txt
```

- The main prerequisite for this code to run is installing nodejs on the machine . In order to run the code go to the root folder of the code in linux / macOS terminal or command prompt and type the command :

```
make clean install build deploy result
```