2020 IRTM HW2 Report

Author

B06705023 資管四 邱廷翔

Environment

- python >= 3.6
- Linux >= 16.04

Requirments

- nltk

To install the required libraries, run the following command.

```
pip install -r requirements.txt
```

Executing the code

```
python main.py doc1 doc2 # e.g. python main.py 1 8
```

output

```
reading all documents...
writing vector files...
cosine sim for doc 1 and 8: [[0.26809512]]
```

- Before running the code, make sure the directory *IRTM* is present.
- Output will be saved in *dictionary.txt* and *tfidf* folder.

Program descriptions

The program can be broke into several phases.

1. Traverse the folder
    - For each file:
        1. Use *tokenization*, which is from HW1, to preprocess each line of the document.
        2. After we acquired a set of tokens used in the document, update the *doc_freq* dictionary.
        3. *doc_freq* is a dictionary with *key* being a corpus and *value* being the term's document-wise frequency.
2. Write the *doc_freq* to file and name it *dictionary.txt*.
3. Calculate the *idf* dictionary given *doc_freq* dictionary.
    - $idf(t) = \log_{10}[\frac{N+1}{df(t)+1}] + 1$
4. For each file:
    1. Use *tokenization* to tokenize the whole document.

2. For each term, calculate its *tf-idf* score.
3. Generate a vector with size equal to the dictioanry size, and fill the tfidf scores into the corresponding position of corpus.

Cosine Similarity

1. Read the two arguments (*doc1*, *doc2*) from the command line.
2. Read the vector file from *doc1.txt*, *doc2.txt*.
3. Construct the tfidf vector from file respectively.
    1. Fill the value into the tfidf file given in the text vector file.
    2. Others remain zero if not mentioned.
4. Compute cosine similarity provided by scikit learn.