ReadMe for TF-IDF AWS MapReduce

## File List

mr\_per\_term\_idf.py -- calculate idf score for each term in the docs

mr\_tfidf\_aws.py -- calculate tf-idf for each (term,doc) pair

boto\_demo.py -- the mail file to run above two files

## Config before running

### 1. config mr\_per\_term\_idf.py

Reason for this: It’s not worth overengineering a simple task which is calculated using simple command line in 1s, running it on aws makes the loss outweighs the gain.

(a) get the total number of lines TotalNumberofDocs for all the files, which is easily got by using command “wc -l files”

(b) manually modify line 27

|  |
| --- |
| idf = math.log10(TotalNumberofDocs \* 1.0 / sum(howmany)) |

replace TotalNumberofDocs with the number get from step (a).

Backup plan 1: TotalNumberofDocs can be get from the input args while running python mr\_per\_terms.py –r emr inputfiles args1.

Backup plan 2: TotalNumberofDocs can also be get by running class MRDocNum.run() in the main class, and store the result in somewhere on aws bucket, and then readin this result on the next step for calculating idf.

(Pity I’m not able to test it since my permission to those buckets has been revoked already).

### 2. config boto\_demo.py

(a) manully modify line 23

|  |
| --- |
| out\_text = subprocess.Popen(['python', 'mr\_per\_term\_idf.py','-r','emr','dataset/arts\_after\_stemming.nlp'], shell=False, stdout=subprocess.PIPE).communicate()[0] |

replace the red marked text with the your input files.

(b) manully modify line 39 -- similar with step2(a).

## How to run

$ python boto\_demo.py

tf-idf result will show up on terminal, one can also save tf-idf result in file.txt by running the following cmd:

$ python boto\_demo.py > file.txt

Input can be several files at the same time, for excemple,

$ python boto\_demo.py doc1.txt doc2.txt doc3.txt …

## Result

|  |
| --- |
| ...  {"term\_docId": {"term": "zur", "docId": 136}, "tfidf": 3.4646385590950328}  {"term\_docId": {"term": "zur", "docId": 2541}, "tfidf": 3.4646385590950328}  {"term\_docId": {"term": "zur", "docId": 97}, "tfidf": 3.4646385590950328}  {"term\_docId": {"term": "zutterm", "docId": 1471}, "tfidf": 3.9417598138146954}  {"term\_docId": {"term": "zwart", "docId": 438}, "tfidf": 3.9417598138146954}  {"term\_docId": {"term": "zwed", "docId": 1379}, "tfidf": 3.9417598138146954}  {"term\_docId": {"term": "zwick", "docId": 739}, "tfidf": 3.9417598138146954}  {"term\_docId": {"term": "zydeco", "docId": 1472}, "tfidf": 3.6407298181507142}  {"term\_docId": {"term": "zydeco", "docId": 1578}, "tfidf": 7.2814596363014283}  ... |

Notify: The intermidiate data is stored under folder test1 in bucket “emr.datascience.testing” on amazon s3, if the bucket is deleted for some reason in the future, please modify the location related lines in file “boto\_demo.py” and “mr\_tfidf\_aws.py”.