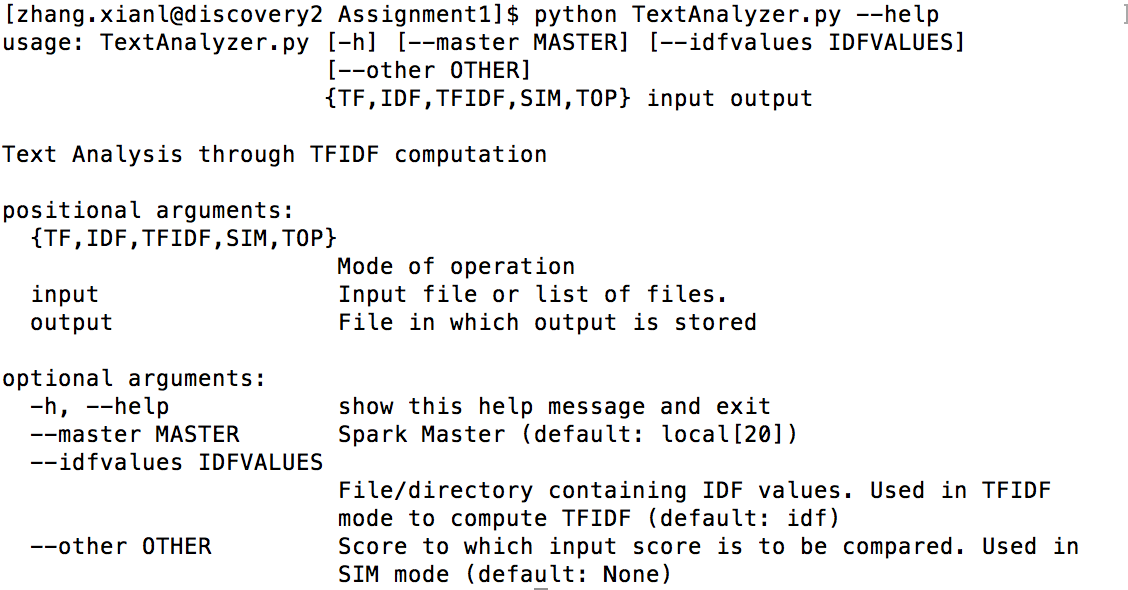
**Question 0:**

**(a)**

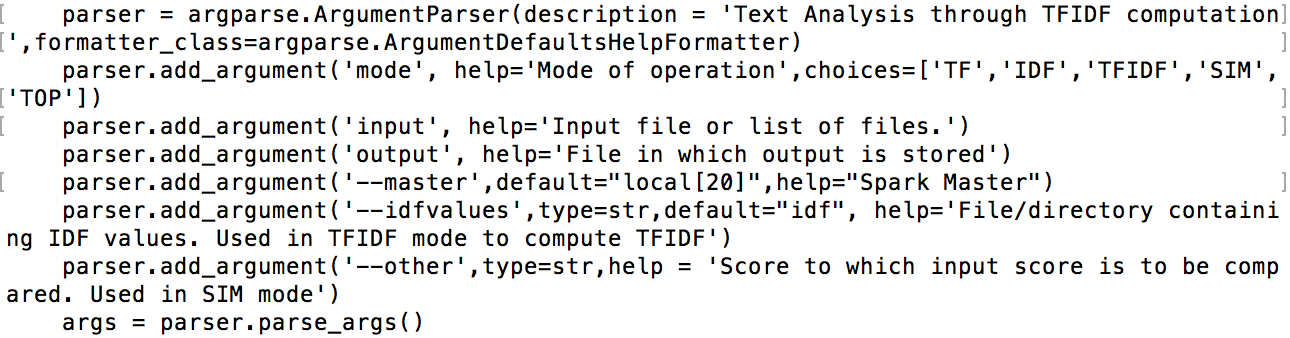
The argparse module generates help and usage messages, indicates input/output files as well as figure out how to parse those out of sys.argv.

**(b)**



Figure

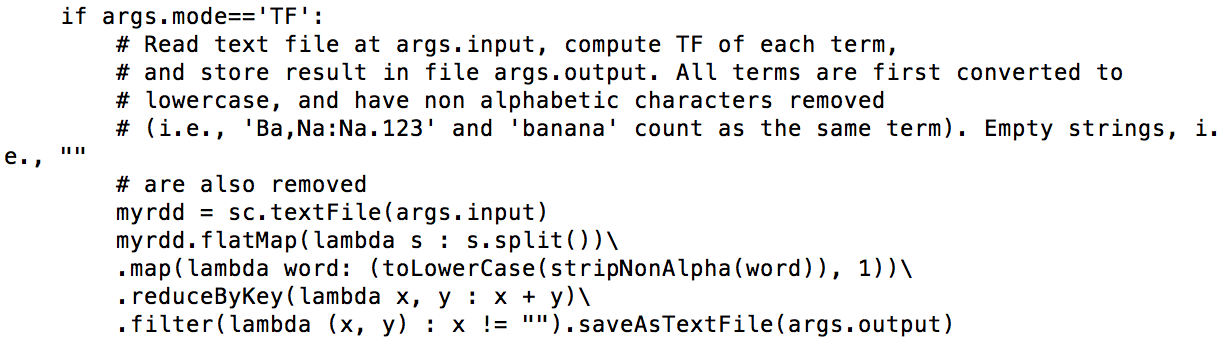
Figure 1 shows what the “python TextAnalyzer.py --help” print, the following lines cause this printed:



**Question 1:**

**(a)**

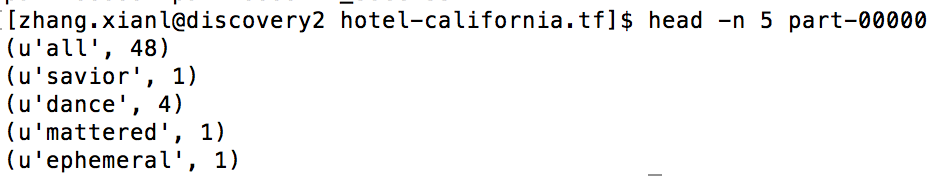
The modified code is as following:

****

First we use sc.textFile() to make the article as rdd, and then the flatMap() makes the article as a tuple of all words. Map() is for giving counter to each word, in which we also call toLowerCase() and stripNonAlpha() to standardize words as instruction said. By using reduceBykey(), we reduce all same words’ counters into one structure. And at last, we use filter() to remove the key empty string ("") and save results in output file.

**(b)**

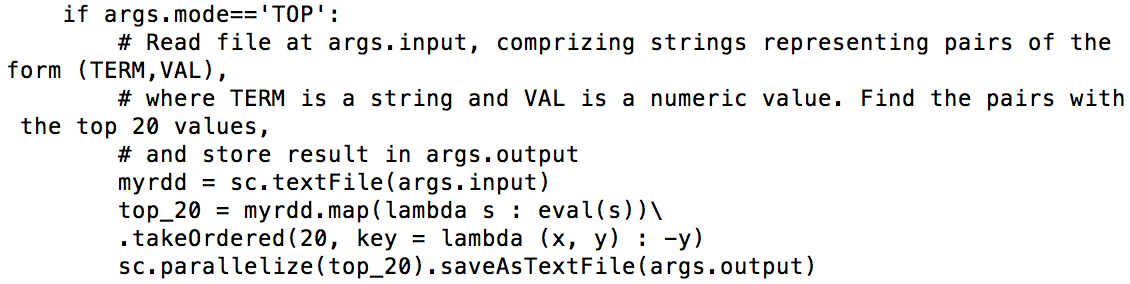
This directory contains three files, part-00000, part-00001 and \_SUCCESS. The first 5 lines of file part-00000 is as following:

****

**Question 2:**

**(a)**

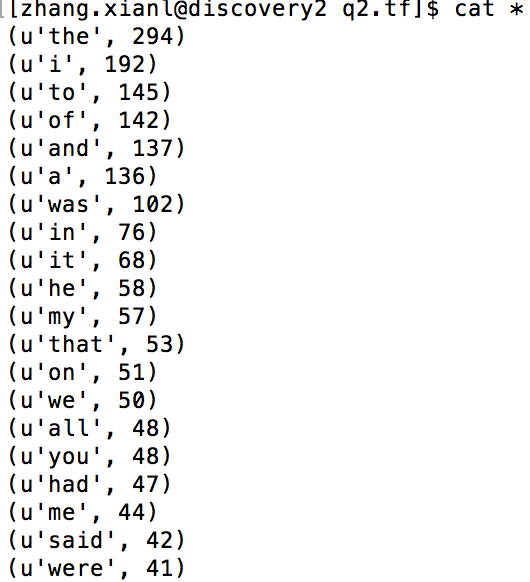
The following is the code I wrote:

****

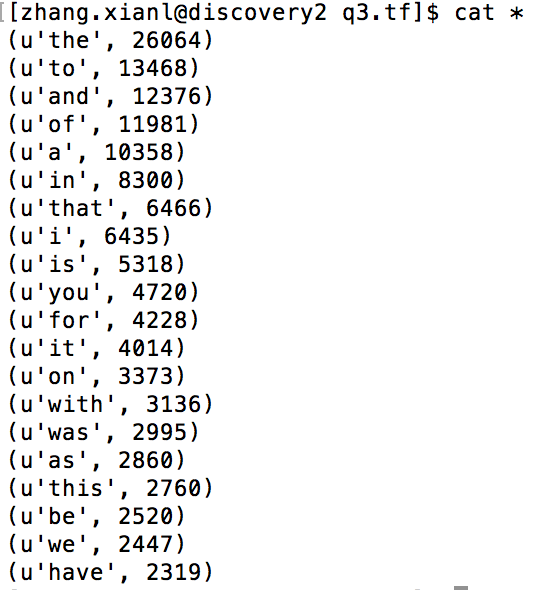
myrdd creates rrd from args.input, and top\_20 makes data from string to tuple through eval() and sorts data by takeOrdered(). At last use sc.parallelize() to make it as rdd, and save it in args.out file.

**(b)**

The most 20 frequent words are as following:

****

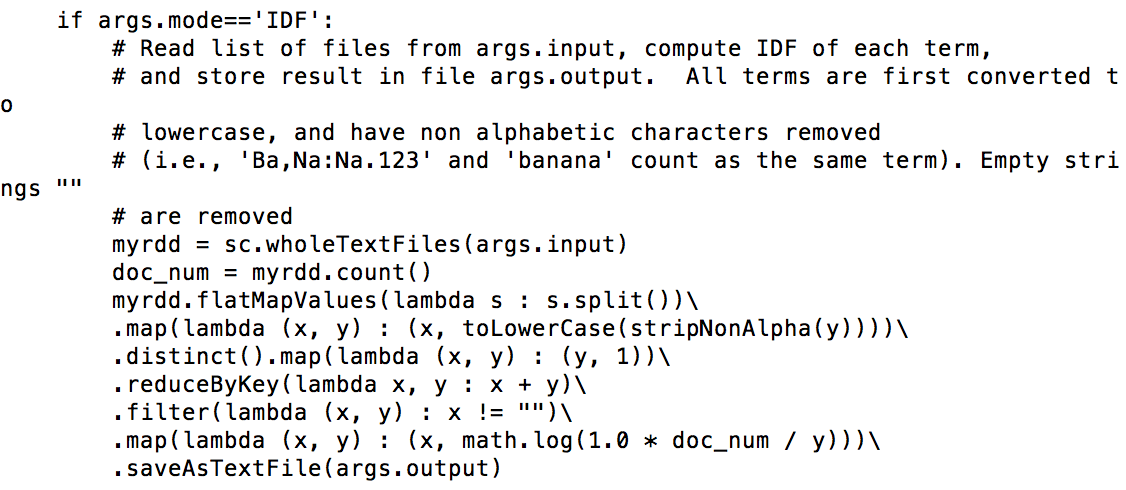
**Question 3:**

****

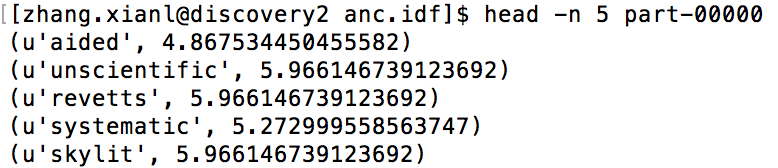
**Question 4:**

**(a)**

The code I modified is as following:

****

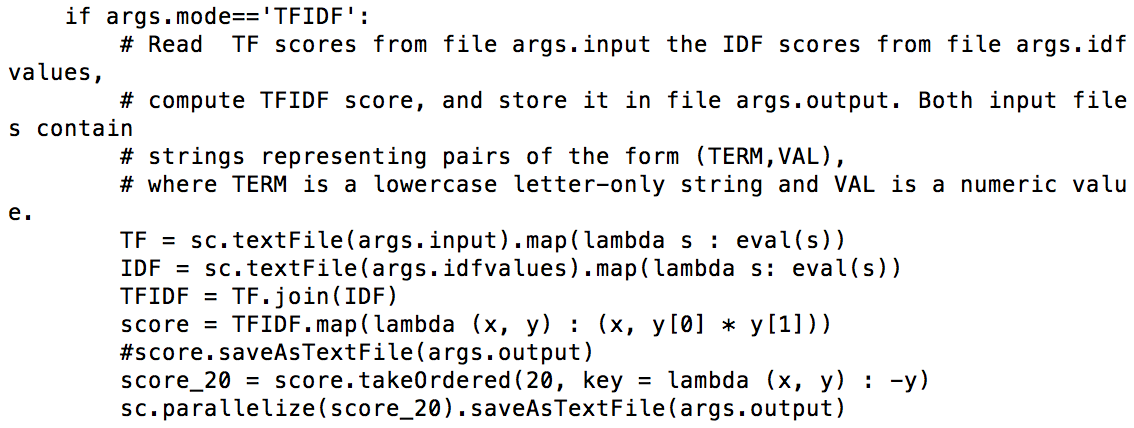
**(b)**

****

**Question 5:**

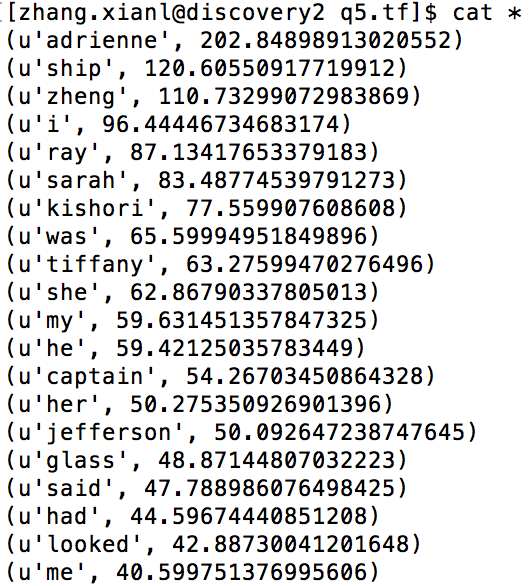
**(a)**

The modified code is as following:

****

**(b)**

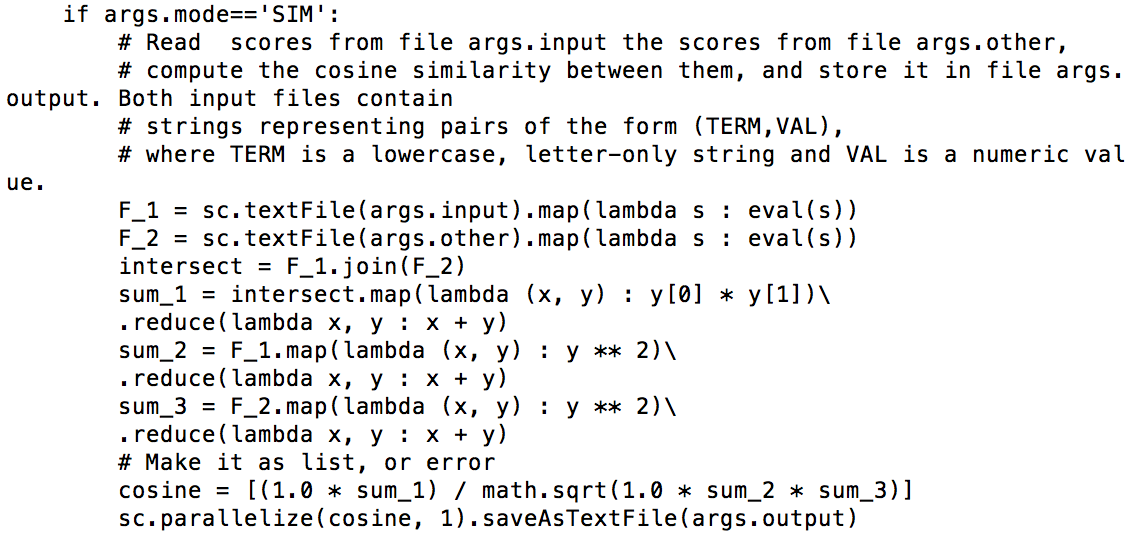
The top 20 TFIDF files are as following, they are different from Q2(b). Apparently, TFIDF is more representative than TF, since it considers the correlation between documents and terms. TF only takes the term frequency into account, but some words may be just frequent in one or two articles, which may affect the results we want.



**Question 6:**

**(a)**

First, use command “find masc\_500k\_texts -name filename -type d” to find path of these directory, and calculate parameters we need. The modified code is as following:



**(b)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | face-to-face | fiction | spam |
| face-to-face | 1 | 0.285739757614 | 0.218984526266 |
| fiction | 0.285739757614 | 1 | 0.313996507076 |
| spam | 0.218984526266 | 0.313996507076 | 1 |