Tommy Baird

605.744 Information Retrieval

Programming Assignment #1

Normalization

To normalize the text, I first started by removing characters that I did not see as relevant. This was mostly made up of our normal universe of punctuation. This was split into ones that I wanted to replace with a blank string as if it was never there and ones that I replaced with a space. Then, we set all remaining characters to lower case and split by a remaining universe of punctuation characters. The ones that are replaced with a space is not much different than splitting by those values, but I split them into two sets as I ran into some odd characters that I wanted to treat special as the computer read them in an odd fashion. Note "â€" as a value that I replaced with a space which is how the computer chose to read a "—" character.

As I iterated through test cases, I played around with different encoding options. The rfa.txt only worked with utf-8, so that is what I settled on. As mentioned, it led to some odd results in how it chose to read in different punctuation characters, so I had to add more values to split by. For example, it saw a difference between ‘, `, and ’, so I made sure to account for all three. Cases like these will probably arise more and more as we iterate through the semester, but we will look to fine tune our logic as the complexity of our programs grows.

The full list of punctuations that we handled is included in the code that is attached on later pages of this document.

Output Summary

The main similarity between the two outputs is the number of stop words that make up the top 100 words. We must scroll down to rank 25 or 30 in each before we would really be able to decipher the difference between the two.

I was surprised at the percentage of words that fell into our "hapex legomena" category in the RFA sample versus the Sense sample. I figured that we would see our stop words control that percentage in a much larger sample of documents/words. However, there is also the extra effect of a wider range of writers and topics that are covered in those news snippets than a single novel from a single author which could constitute more variety from the RFA sample.

Output 1:

This is our output for sense.txt

We have processed 1862 total paragraphs
We have found a vocabulary size of 6911
We have found a collection size of 119958

1: 'to' has a collection frequency of 4115 and document frequency of 1203
2: 'the' has a collection frequency of 4104 and document frequency of 1138
3: 'of' has a collection frequency of 3569 and document frequency of 1097
4: 'and' has a collection frequency of 3491 and document frequency of 1166
5: 'her' has a collection frequency of 2528 and document frequency of 809
6: 'a' has a collection frequency of 2092 and document frequency of 930
7: 'i' has a collection frequency of 1997 and document frequency of 634
8: 'in' has a collection frequency of 1979 and document frequency of 913
9: 'was' has a collection frequency of 1857 and document frequency of 774
10: 'it' has a collection frequency of 1720 and document frequency of 840
11: 'she' has a collection frequency of 1610 and document frequency of 700
12: 'that' has a collection frequency of 1377 and document frequency of 769
13: 'be' has a collection frequency of 1291 and document frequency of 715
14: 'for' has a collection frequency of 1261 and document frequency of 702
15: 'not' has a collection frequency of 1245 and document frequency of 770
16: 'as' has a collection frequency of 1221 and document frequency of 625
17: 'you' has a collection frequency of 1169 and document frequency of 553
18: 'he' has a collection frequency of 1104 and document frequency of 540
19: 'his' has a collection frequency of 1020 and document frequency of 454
20: 'had' has a collection frequency of 998 and document frequency of 533
21: 'with' has a collection frequency of 992 and document frequency of 613
22: 'but' has a collection frequency of 885 and document frequency of 636
23: 'at' has a collection frequency of 838 and document frequency of 559
24: 'have' has a collection frequency of 818 and document frequency of 489
25: 'by' has a collection frequency of 749 and document frequency of 489
26: 'is' has a collection frequency of 745 and document frequency of 463
27: 'on' has a collection frequency of 694 and document frequency of 470
28: 'all' has a collection frequency of 652 and document frequency of 453
29: 'so' has a collection frequency of 635 and document frequency of 421
30: 'my' has a collection frequency of 628 and document frequency of 311
31: 'him' has a collection frequency of 626 and document frequency of 353
32: 'elinor' has a collection frequency of 615 and document frequency of 540
33: 'which' has a collection frequency of 593 and document frequency of 391
34: 'could' has a collection frequency of 578 and document frequency of 404
35: 'no' has a collection frequency of 567 and document frequency of 405
36: 'from' has a collection frequency of 538 and document frequency of 371
37: 'mrs' has a collection frequency of 530 and document frequency of 396

38: 'they' has a collection frequency of 518 and document frequency of 335
39: 'would' has a collection frequency of 513 and document frequency of 351
40: 'very' has a collection frequency of 500 and document frequency of 388
41: 'their' has a collection frequency of 496 and document frequency of 307
42: 'marianne' has a collection frequency of 484 and document frequency of 406
43: 'them' has a collection frequency of 465 and document frequency of 324
44: 'been' has a collection frequency of 440 and document frequency of 314
45: 'were' has a collection frequency of 440 and document frequency of 319
46: 'what' has a collection frequency of 435 and document frequency of 313
47: 'this' has a collection frequency of 432 and document frequency of 349
48: 'me' has a collection frequency of 421 and document frequency of 229
49: 'more' has a collection frequency of 406 and document frequency of 317
50: 'said' has a collection frequency of 397 and document frequency of 369
51: 'any' has a collection frequency of 390 and document frequency of 310
52: 'your' has a collection frequency of 385 and document frequency of 249
53: 'every' has a collection frequency of 377 and document frequency of 279
54: 'will' has a collection frequency of 363 and document frequency of 230
55: 'than' has a collection frequency of 360 and document frequency of 284
56: 'such' has a collection frequency of 359 and document frequency of 283
57: 'or' has a collection frequency of 356 and document frequency of 262
58: 'an' has a collection frequency of 344 and document frequency of 275
59: 'do' has a collection frequency of 320 and document frequency of 270
60: 'one' has a collection frequency of 318 and document frequency of 259
61: 'when' has a collection frequency of 306 and document frequency of 257
62: 'if' has a collection frequency of 293 and document frequency of 246
63: 'much' has a collection frequency of 288 and document frequency of 245
64: 'only' has a collection frequency of 287 and document frequency of 240
65: 'must' has a collection frequency of 283 and document frequency of 228
66: 'own' has a collection frequency of 271 and document frequency of 218
67: 'who' has a collection frequency of 268 and document frequency of 213
68: 'herself' has a collection frequency of 253 and document frequency of 208
69: 'did' has a collection frequency of 246 and document frequency of 203
70: 'now' has a collection frequency of 237 and document frequency of 207
71: 'time' has a collection frequency of 237 and document frequency of 204
72: 'should' has a collection frequency of 236 and document frequency of 187
73: 'am' has a collection frequency of 236 and document frequency of 179
74: 'how' has a collection frequency of 235 and document frequency of 182
75: 'there' has a collection frequency of 235 and document frequency of 197
76: 'well' has a collection frequency of 232 and document frequency of 196
77: 'are' has a collection frequency of 232 and document frequency of 184
78: 'know' has a collection frequency of 231 and document frequency of 187
79: 'sister' has a collection frequency of 226 and document frequency of 198
80: 'dashwood' has a collection frequency of 218 and document frequency of 190
81: 'though' has a collection frequency of 216 and document frequency of 192

82: 'some' has a collection frequency of 215 and document frequency of 184
83: 'we' has a collection frequency of 215 and document frequency of 152
84: 'might' has a collection frequency of 215 and document frequency of 174
85: 'has' has a collection frequency of 213 and document frequency of 153
86: 'think' has a collection frequency of 210 and document frequency of 189
87: 'miss' has a collection frequency of 210 and document frequency of 167
88: 'mother' has a collection frequency of 210 and document frequency of 181
89: 'can' has a collection frequency of 209 and document frequency of 174
90: 'edward' has a collection frequency of 207 and document frequency of 167
91: 'jennings' has a collection frequency of 204 and document frequency of 173
92: 'after' has a collection frequency of 203 and document frequency of 191
93: 'before' has a collection frequency of 199 and document frequency of 174
94: 'never' has a collection frequency of 189 and document frequency of 168
95: 'nothing' has a collection frequency of 188 and document frequency of 159
96: 'other' has a collection frequency of 182 and document frequency of 159
97: 'too' has a collection frequency of 181 and document frequency of 145
98: 'soon' has a collection frequency of 179 and document frequency of 161
99: 'mr' has a collection frequency of 178 and document frequency of 152
100: 'good' has a collection frequency of 177 and document frequency of 154
500: 'misery' has a collection frequency of 28 and document frequency of 26
1000: 'consciousness' has a collection frequency of 12 and document frequency of 12
5000: 'conform' has a collection frequency of 1 and document frequency of 1

The number of words that only appeared in one document are: 2900
The percentage of these terms relative to the full dictionary is: 41.962%

Output 2:

This is our output for rfa.txt

We have processed 99999 total paragraphs
We have found a vocabulary size of 120300
We have found a collection size of 6062430

1: 'the' has a collection frequency of 353669 and document frequency of 74215
2: 'to' has a collection frequency of 180878 and document frequency of 56137
3: 'of' has a collection frequency of 170086 and document frequency of 53440
4: 'in' has a collection frequency of 141639 and document frequency of 51331
5: 'and' has a collection frequency of 141420 and document frequency of 48079
6: 'a' has a collection frequency of 115014 and document frequency of 42906
7: 'that' has a collection frequency of 68325 and document frequency of 30071
8: 'for' has a collection frequency of 64040 and document frequency of 29832
9: 'said' has a collection frequency of 60562 and document frequency of 32130
10: 'on' has a collection frequency of 57690 and document frequency of 26640
11: 'by' has a collection frequency of 50188 and document frequency of 21019
12: 'is' has a collection frequency of 43561 and document frequency of 21239
13: 'he' has a collection frequency of 35277 and document frequency of 17155
14: 'with' has a collection frequency of 33643 and document frequency of 18984
15: 'have' has a collection frequency of 32052 and document frequency of 17175
16: 'was' has a collection frequency of 31929 and document frequency of 16096
17: 'as' has a collection frequency of 31928 and document frequency of 16825
18: 'from' has a collection frequency of 30558 and document frequency of 17285
19: 'are' has a collection frequency of 28312 and document frequency of 15453
20: 'has' has a collection frequency of 27428 and document frequency of 15544
21: 'they' has a collection frequency of 27136 and document frequency of 13509
22: 'it' has a collection frequency of 27051 and document frequency of 15354
23: 'at' has a collection frequency of 26883 and document frequency of 15918
24: 'be' has a collection frequency of 24711 and document frequency of 14123
25: 'china' has a collection frequency of 24574 and document frequency of 15548
26: 'had' has a collection frequency of 21045 and document frequency of 11065
27: 'government' has a collection frequency of 20908 and document frequency of 12724
28: 'an' has a collection frequency of 20900 and document frequency of 13303
29: 'chinese' has a collection frequency of 20044 and document frequency of 11956
30: 'but' has a collection frequency of 19768 and document frequency of 13963
31: 'his' has a collection frequency of 19450 and document frequency of 9707
32: 'who' has a collection frequency of 19381 and document frequency of 11558
33: 'been' has a collection frequency of 18966 and document frequency of 11492
34: 'their' has a collection frequency of 18678 and document frequency of 10973
35: 'its' has a collection frequency of 18607 and document frequency of 11214
36: 'were' has a collection frequency of 18197 and document frequency of 10677
37: 'will' has a collection frequency of 17163 and document frequency of 9805

38: 'not' has a collection frequency of 16980 and document frequency of 10935
39: 'this' has a collection frequency of 16402 and document frequency of 10854
40: 'people' has a collection frequency of 15321 and document frequency of 9380
41: 'i' has a collection frequency of 15271 and document frequency of 7482
42: 'which' has a collection frequency of 14501 and document frequency of 9270
43: 'us' has a collection frequency of 14298 and document frequency of 8534
44: 'rights' has a collection frequency of 14228 and document frequency of 7625
45: 'or' has a collection frequency of 13625 and document frequency of 8855
46: 'party' has a collection frequency of 13608 and document frequency of 7436
47: 'after' has a collection frequency of 13134 and document frequency of 8985
48: 'we' has a collection frequency of 13095 and document frequency of 7219
49: 'also' has a collection frequency of 12263 and document frequency of 8696
50: 'would' has a collection frequency of 12018 and document frequency of 7648
51: 'chinas' has a collection frequency of 12007 and document frequency of 8910
52: 'police' has a collection frequency of 11955 and document frequency of 6445
53: 'more' has a collection frequency of 11911 and document frequency of 8387
54: 'told' has a collection frequency of 11371 and document frequency of 8445
55: 'about' has a collection frequency of 11298 and document frequency of 8006
56: 'year' has a collection frequency of 11148 and document frequency of 7746
57: 'cambodia' has a collection frequency of 10473 and document frequency of 5974
58: 'service' has a collection frequency of 10464 and document frequency of 7493
59: 'up' has a collection frequency of 9799 and document frequency of 7209
60: 'all' has a collection frequency of 9555 and document frequency of 6918
61: 'them' has a collection frequency of 9380 and document frequency of 6538
62: 'one' has a collection frequency of 9350 and document frequency of 7051
63: 'over' has a collection frequency of 9347 and document frequency of 6930
64: 'authorities' has a collection frequency of 9342 and document frequency of 6412
65: 'there' has a collection frequency of 9314 and document frequency of 6521
66: 'political' has a collection frequency of 9174 and document frequency of 5899
67: 'she' has a collection frequency of 9037 and document frequency of 4934
68: 'national' has a collection frequency of 8790 and document frequency of 5792
69: 'no' has a collection frequency of 8655 and document frequency of 6311
70: 'two' has a collection frequency of 8572 and document frequency of 6271
71: 'out' has a collection frequency of 8433 and document frequency of 6314
72: 'than' has a collection frequency of 8390 and document frequency of 6356
73: 'new' has a collection frequency of 8335 and document frequency of 5940
74: 'if' has a collection frequency of 8299 and document frequency of 6122
75: 'rfa' has a collection frequency of 8259 and document frequency of 6338
76: 'years' has a collection frequency of 8153 and document frequency of 6261
77: 'last' has a collection frequency of 8017 and document frequency of 6096
78: 'hong' has a collection frequency of 7917 and document frequency of 3746
79: 'percent' has a collection frequency of 7900 and document frequency of 3890
80: 'rfas' has a collection frequency of 7865 and document frequency of 6641
81: 'cambodian' has a collection frequency of 7858 and document frequency of 5018

82: 'so' has a collection frequency of 7827 and document frequency of 5980
83: 'officials' has a collection frequency of 7777 and document frequency of 5669
84: 'other' has a collection frequency of 7757 and document frequency of 6050
85: 'some' has a collection frequency of 7713 and document frequency of 5973
86: 'beijing' has a collection frequency of 7594 and document frequency of 5263
87: 'state' has a collection frequency of 7503 and document frequency of 5346
88: 'against' has a collection frequency of 7448 and document frequency of 5300
89: 'official' has a collection frequency of 7340 and document frequency of 5389
90: 'hun' has a collection frequency of 7313 and document frequency of 4115
91: 'her' has a collection frequency of 7231 and document frequency of 3446
92: 'local' has a collection frequency of 7192 and document frequency of 4736
93: 'human' has a collection frequency of 7172 and document frequency of 4575
94: 'under' has a collection frequency of 7024 and document frequency of 5269
95: 'any' has a collection frequency of 6885 and document frequency of 5108
96: 'kong' has a collection frequency of 6881 and document frequency of 3505
97: 'may' has a collection frequency of 6875 and document frequency of 5059
98: 'when' has a collection frequency of 6856 and document frequency of 5562
99: 'because' has a collection frequency of 6825 and document frequency of 5426
100: 'now' has a collection frequency of 6814 and document frequency of 5314
500: 'deputy' has a collection frequency of 1531 and document frequency of 1322
1000: 'nation' has a collection frequency of 814 and document frequency of 734
5000: 'makers' has a collection frequency of 99 and document frequency of 93

The number of words that only appeared in one document are: 72898
The percentage of these terms relative to the full dictionary is: 60.597%

```python
# Press the green button in the gutter to run the script.
import re
import pandas as pd
import sys

def findIndex(rawValue):
    # knowing standard format for ID, we can remove the irrelevant characters
    valuesToRemove = ['id=', '>']
    indexValue = rawValue
    for currentRemoveValue in valuesToRemove:
        indexValue = indexValue.replace(currentRemoveValue, '')

    # return the remaining value as an integer
    return int(indexValue)


def removeUselessChar(rawValue):
    # below are characters that we want to remove from strings
    newValue = rawValue
    # these values we want to remove with empty string
    valuesToRemove = ['"', "'", "`", "\n", "â€™", "{", "}", "[", "]", "#", "$", "’", "."
, ";", "”"]
    # these we want to remove but replace with a blank space
    valuesToSpaceReplace = ["\t", "â€”", "&", "~"]

    # iterate through both lists to update our value
    for currentRemoveValue in valuesToRemove:
        newValue = newValue.replace(currentRemoveValue, '')
    for currentNewSpace in valuesToSpaceReplace:
        newValue = newValue.replace(currentNewSpace, ' ')

    return newValue


def addValues(currentTable, newValues, currentDocID):
    # create new row for our output dataframe
    newRows = pd.DataFrame({
        'DocID': [currentDocID] * len(newValues),  # Repeat the name for each city
        'Value': newValues
    })

    # add to existing dataframe and return the updated table
    currentTable = pd.concat([currentTable, newRows], ignore_index=True)
    return currentTable


if __name__ == '__main__':
    # init our variables needed to track reading of
    fullLog = pd.DataFrame()
    currentID = 0

    # read in the file name which is passed as a command line arg
    currentFileName = sys.argv[1]
    print(f"Reading in text file: {currentFileName}")

    # create our output file to store our results
    outputFileName = currentFileName.replace(".txt", "Output.txt")
    outputFile = open(outputFileName, "w")
    # write opening line of document
    outputFile.write(f"This is our output for {currentFileName}\n\n")

    # iterate through the open file line by line
    with open(currentFileName, 'r', encoding='utf-8') as fullFile:
        for nextLine in fullFile:
            # remove characters that we do not need
            cleanLine = removeUselessChar(nextLine)
            # empty lines and end of paragraphs we can skip
            if cleanLine == '' or cleanLine == '</P>':
                pass
```

```
            else:
                # set all to lower case
                cleanLine = cleanLine.lower()
                # these values we want to split into new word as they are natural breaks in a sentence or a paragraph
                cleanLine = re.split(r'[-_|:/\s,()-]', cleanLine)
                # check if beginning of new paragraph where we will extract our paragraph id
                if cleanLine[0] == '<p':
                    newIndex = findIndex(cleanLine[1])
                    currentID = newIndex
                    print(f"Processing document {newIndex}")
                # otherwise it is a good line to process for our tokens
                else:
                    fullLog = addValues(fullLog, cleanLine, currentID)

    # remove any stray empty values
    fullLog = fullLog[fullLog['Value'] != '']

    # find the summary of number of paragraphs, unique words, and total words
    numParagraphs = fullLog['DocID'].nunique()
    numUniqueWords = fullLog['Value'].nunique()
    numTotalWords = len(fullLog)

    # write our initial summary items
    outputFile.write(f"We have processed {numParagraphs} total paragraphs\n")
    outputFile.write(f"We have found a vocabulary size of {numUniqueWords}\n")
    outputFile.write(f"We have found a collection size of {numTotalWords}\n\n")

    # group the words by collection frequency
    collectionFreq = fullLog.groupby('Value').agg({'DocID': 'count'})
    collectionFreq = collectionFreq.sort_values(by='DocID', ascending=False)
    # rename the count column
    collectionFreq = collectionFreq.rename(columns={'DocID': 'ColFreq'})

    # group the words by document frequency
    # first drop duplicates that represent multiple instances of word in document
    documentFreq = fullLog.drop_duplicates()
    # then find count of each value which now represents the number of documents it appears in
    documentFreq = documentFreq.groupby('Value').agg({'DocID': 'count'})
    documentFreq = documentFreq.sort_values(by='DocID', ascending=False)
    # rename the count column
    documentFreq = documentFreq.rename(columns={'DocID': 'DocFreq'})

    # join together the collection and document freq
    collectionFreq = collectionFreq.join(documentFreq).reset_index()

    # print out information of top 50 records
    for currentRank in range(100):
        currentRankID = currentRank + 1
        currentWord = collectionFreq.loc[currentRank, 'Value']
        currentColFreq = collectionFreq.loc[currentRank, 'ColFreq']
        currentDocFreq = collectionFreq.loc[currentRank, 'DocFreq']

        outputFile.write(f"{currentRankID}: '{currentWord}' has a collection frequency of {currentColFreq} and document frequency of {currentDocFreq}\n")

    # print info for words 500, 1000, and 1500
    otherIndices = [500, 1000, 5000]
    for currentRankID in otherIndices:
        currentRank = currentRankID - 1
        currentWord = collectionFreq.loc[currentRank, 'Value']
        currentColFreq = collectionFreq.loc[currentRank, 'ColFreq']
        currentDocFreq = collectionFreq.loc[currentRank, 'DocFreq']

        outputFile.write(f"{currentRankID}: '{currentWord}' has a collection frequency of {currentColFreq} and document frequency of {currentDocFreq}\n")

```

```
    # find words only in one document
    singleDocWords = documentFreq[documentFreq['DocFreq'] == 1].reset_index()
    numSingleDocWords = len(singleDocWords)
    percentSingleDocWord = round((numSingleDocWords / numUniqueWords) * 100, 3)
    outputFile.write(f"\nThe number of words that only appeared in one document are: {numSingleDocWords}")
    outputFile.write(f"\nThe percentage of these terms relative to the full dictionary is: {percentSingleDocWord}%")

```