Module 1 Programing Assignment

Max Robinson

September 4, 2018

1 Summary

In this assignment two document sets, "Yelp" and "Headlines" were processed to produce a summary of the document set. The summary for each document set consists of the number of total words encountered, the number of unique words observed, the total number of documents processed, the top 100 most frequent terms, the 500th 1,00th, and 5,000th frequent terms, and the number and percentage of words that occur in only one document.

The document sets were contained in single text files where each document in the file was identified by surrounding text marking the stop and start of the document, and providing the document ID.

When processing the documents the text was broken into words and normalized. Words were determined by splitting the text on spaces. Internal punctuation was left intact. The words were then normalized by lower-casing all words, and removing excess punctuation from the front or back of a word. Any word that was itself only punctuation was also disregarded. The set of characters characterized as punctuation are as follows: '.', ',', "'", '"', '?', '!', ':', ';', '(', ')', '[', ']', '', '', '&', '—', '-'.

For the two document sets there are a few similarities and differences in the results. Both document sets have fairly similar words in the top 100 words. Many of the words in common such as 'the', 'and', 'of', are common short articles or shorter common words like 'we', 'they', 'as' that are typical in the English language.

Longer words in the top 100 start to very based on the content of the document set. For instance the Headlines document set has words such as 'global' or '2015' in the top 100 where Yelp has words like 'food' or 'service'. These differences start to make sense when the context of the documents is taken into account.

2 Code

The following subsections provide the code that was created to process and produce the results for this assignment.

2.1 Module1.py

The main program for the assignment.

```
from builtins import sorted

from ingest import FileIngester


class Module1:

    def __init__(self, file: str) -> None:
        self.file = file
        self.report = {
            "num_of_paragraphs": -1,
            "num_unique_words": -1,
            "num_total_words": -1,
        }

        """
        {
            word: {"col_freq": int, "doc_freq": int}
        }
        """
        self.tf = {}
        self.punct = ['.',  ',', '"', "'", '?', '!', ':', ';', '(', ')',
            '[', ']', '{', '}', '&', '|', '-']
        self.words = {}

    def run(self) -> None:
        ingester = FileIngester(self.file)
        docs = ingester.read()

        self.report["num_of_paragraphs"] = len(docs)

        clean_docs = self.clean_docs(docs)
        self.process_docs(clean_docs)
        # print(self.tf)
        self.report["num_unique_words"] = len(self.tf.keys())

        print(self.report)

        words = sorted(self.tf.items(), key=self.tf_compare, reverse=True)

        one_doc_words = self.calc_number_of_words_in_one_doc()

        percentage_one_doc_words = one_doc_words/ self.report["num_unique_words"]
```

```
        print("Number of words in only 1 document: {}"
                    .format(one_doc_words))
        print("Percentage of words in only 1 document: {}"
                    .format(percentage_one_doc_words))

        print("Top 100")
        self.print_results(words[0:100])
        print("Top 500th word")
        self.print_results(words[499:500])
        print("Top 1000th word")
        self.print_results(words[999:1000])
        print("Top 5000th word")
        self.print_results(words[4999:5000])

    def tf_compare(self, item: tuple):
        """
        item = (key, dict)
        :param item:
        :return:
        """

        return item[1]["col_freq"]

    def clean_docs(self, docs: list):
        new_docs = []
        for doc in docs:
            new_doc = {
                "id": doc["id"],
                "words": []
            }

            content = doc["content"]  # type: str
            words = [word.lower() for word in content.split() if
                    word.lower() not in self.punct]  # type: list
            words = self.strip_punct(words)  # type: list

            new_doc["words"] = words

            new_docs.append(new_doc)

        return new_docs

    def strip_punct(self, words: list) -> list:
        for punct in self.punct:
            words = [word.strip(punct) for word in words]
```

```
        return words

    def process_docs(self, docs):
        for doc in docs:
            doc_set = {}
            for word in doc["words"]:
                self.report["num_total_words"] += 1

                if word in doc_set:
                    doc_set[word] += 1
                else:
                    doc_set[word] = 1

            for word in doc_set:
                if word in self.tf:
                    self.tf[word]["col_freq"] += doc_set[word]
                    self.tf[word]["doc_freq"] += 1
                else:
                    self.tf[word] = {
                        "col_freq": doc_set[word],
                        "doc_freq": 1
                    }

    def print_results(self, list_of_words) -> None:
        """
        list_of_words = [('word', {})]
        :param list_of_words:
        :return:
        """
        for word in list_of_words:
            print("{:>12}: Collection frequency: {:5d}, Document
                frequency:{:5d}".format(word[0], word[1]["col_freq"],
                word[1]["doc_freq"]))
        return

    def calc_number_of_words_in_one_doc(self) -> int:
        total = 0
        for word in self.tf:
            if self.tf[word]["doc_freq"] == 1:
                total += 1

        return total


if __name__ == '__main__':
    m = Module1("yelp.txt")
```

```
    m.run()

    m = Module1("headlines.txt")
    m.run()
```

2.2 ingest.py

A python file dedicated to reading input from a file and parsing the contents into memory.

```
import os


class FileIngester:

    def __init__(self, file_location: str):
        if os.path.isfile(file_location):
            self.file_location = file_location
        else:
            self.file_location = None

        self.documents = []

    def read(self) -> list:
        """
        document = {
            id=number,
            content=""
        }
        :return:
        """
        if self.file_location is None:
            return {}

        lines = []
        with open(self.file_location, 'r') as f:
            document = self.new_document()
            for line in f:
                line = line.strip()

                if line == "":
                    continue

                if "<P ID=" in line:
                    line = line.replace("<P ID=", "")
                    line = line.replace(">", "")
```

```
                    id=-1
                    try:
                        id = int(line)
                        document["id"] = id
                    except:
                        pass

                elif "</P>" in line:
                    self.documents.append(document)
                    document = self.new_document()

                else:
                    document["content"] += line

        return self.documents

    def new_document(self) -> dict:
        return {"id": -1, "content": ""}


if __name__ == '__main__':
    x = FileIngester("yelp.txt")
    docs = x.read()
    print(docs)
```

3 Results

The following sections provide the results for the assignment for each document set.

3.1 Results for Yelp.txt

```
{'num_unique_words': 40527, 'num_total_words': 1262930,
'num_of_paragraphs': 8892}
Number of words in only 1 document: 23962
Percentage of words in only 1 document: 0.5912601475559504

Top 100
         the: Collection frequency:  65253, Document frequency: 8530
         and: Collection frequency:  41322, Document frequency: 8273
           a: Collection frequency:  33449, Document frequency: 7879
           i: Collection frequency:  33264, Document frequency: 7367
          to: Collection frequency:  27960, Document frequency: 7425
         was: Collection frequency:  22223, Document frequency: 5973
          of: Collection frequency:  19541, Document frequency: 6501
```

```
    it: Collection frequency:  17526, Document frequency: 6189
    is: Collection frequency:  15735, Document frequency: 6084
   for: Collection frequency:  14778, Document frequency: 6272
    in: Collection frequency:  13854, Document frequency: 5964
  that: Collection frequency:  11197, Document frequency: 4938
   but: Collection frequency:  10985, Document frequency: 5558
  with: Collection frequency:  10424, Document frequency: 4983
    my: Collection frequency:  10228, Document frequency: 4866
  this: Collection frequency:  10136, Document frequency: 5386
    we: Collection frequency:   9752, Document frequency: 3354
  they: Collection frequency:   8857, Document frequency: 4402
    on: Collection frequency:   8739, Document frequency: 4705
   you: Collection frequency:   8573, Document frequency: 4035
   not: Collection frequency:   8257, Document frequency: 4505
  food: Collection frequency:   7945, Document frequency: 4763
  have: Collection frequency:   7671, Document frequency: 4366
   had: Collection frequency:   7117, Document frequency: 3977
  were: Collection frequency:   6825, Document frequency: 3396
  good: Collection frequency:   6715, Document frequency: 4096
    at: Collection frequency:   6410, Document frequency: 3853
    so: Collection frequency:   6344, Document frequency: 3736
 place: Collection frequency:   6323, Document frequency: 4037
   are: Collection frequency:   5789, Document frequency: 3485
    as: Collection frequency:   5556, Document frequency: 3006
    be: Collection frequency:   5383, Document frequency: 3546
 there: Collection frequency:   4858, Document frequency: 3139
  like: Collection frequency:   4829, Document frequency: 3191
  just: Collection frequency:   4516, Document frequency: 3013
    if: Collection frequency:   4493, Document frequency: 3072
   out: Collection frequency:   4193, Document frequency: 2896
   all: Collection frequency:   4120, Document frequency: 2811
  very: Collection frequency:   4056, Document frequency: 2693
  here: Collection frequency:   4038, Document frequency: 2956
    me: Collection frequency:   4003, Document frequency: 2544
   one: Collection frequency:   3895, Document frequency: 2716
  it's: Collection frequency:   3746, Document frequency: 2387
   our: Collection frequency:   3729, Document frequency: 1932
   get: Collection frequency:   3703, Document frequency: 2632
 their: Collection frequency:   3673, Document frequency: 2323
 great: Collection frequency:   3663, Document frequency: 2613
    or: Collection frequency:   3662, Document frequency: 2543
  when: Collection frequency:   3566, Document frequency: 2542
service: Collection frequency:   3536, Document frequency: 2871
  from: Collection frequency:   3454, Document frequency: 2518
  time: Collection frequency:   3287, Document frequency: 2394
    go: Collection frequency:   3171, Document frequency: 2463
```

```
          up: Collection frequency:   3101, Document frequency: 2243
      really: Collection frequency:   3096, Document frequency: 2144
       would: Collection frequency:   3075, Document frequency: 2150
       which: Collection frequency:   3055, Document frequency: 2105
        some: Collection frequency:   3015, Document frequency: 2166
       about: Collection frequency:   2985, Document frequency: 2201
        back: Collection frequency:   2979, Document frequency: 2295
        what: Collection frequency:   2900, Document frequency: 2154
        been: Collection frequency:   2779, Document frequency: 2127
          an: Collection frequency:   2752, Document frequency: 2106
       order: Collection frequency:   2662, Document frequency: 1809
          no: Collection frequency:   2655, Document frequency: 1916
    ordered: Collection frequency:   2569, Document frequency: 1883
 restaurant: Collection frequency:   2557, Document frequency: 1783
    chicken: Collection frequency:   2543, Document frequency: 1554
       only: Collection frequency:   2476, Document frequency: 1991
       will: Collection frequency:   2447, Document frequency: 1989
       more: Collection frequency:   2397, Document frequency: 1868
        can: Collection frequency:   2357, Document frequency: 1823
       your: Collection frequency:   2341, Document frequency: 1670
      don't: Collection frequency:   2275, Document frequency: 1813
       also: Collection frequency:   2254, Document frequency: 1744
         by: Collection frequency:   2246, Document frequency: 1753
        too: Collection frequency:   2198, Document frequency: 1759
         us: Collection frequency:   2192, Document frequency: 1356
      other: Collection frequency:   2171, Document frequency: 1767
    because: Collection frequency:   2161, Document frequency: 1670
      pizza: Collection frequency:   2133, Document frequency:  837
        got: Collection frequency:   2099, Document frequency: 1543
       even: Collection frequency:   2060, Document frequency: 1650
        i'm: Collection frequency:   2048, Document frequency: 1546
       menu: Collection frequency:   2007, Document frequency: 1514
     little: Collection frequency:   1981, Document frequency: 1527
         he: Collection frequency:   1934, Document frequency:  986
       them: Collection frequency:   1920, Document frequency: 1453
       nice: Collection frequency:   1908, Document frequency: 1509
      after: Collection frequency:   1901, Document frequency: 1493
       i've: Collection frequency:   1856, Document frequency: 1409
       well: Collection frequency:   1853, Document frequency: 1476
       than: Collection frequency:   1844, Document frequency: 1531
        she: Collection frequency:   1841, Document frequency:  860
        has: Collection frequency:   1813, Document frequency: 1495
         do: Collection frequency:   1774, Document frequency: 1397
     pretty: Collection frequency:   1765, Document frequency: 1370
       came: Collection frequency:   1757, Document frequency: 1314
       much: Collection frequency:   1743, Document frequency: 1445
```

```
        best: Collection frequency:   1739, Document frequency: 1455
Top 500th word
        name: Collection frequency:    328, Document frequency:  292
Top 1000th word
         bun: Collection frequency:    132, Document frequency:  109
Top 5000th word
     sincere: Collection frequency:     11, Document frequency:   11
```

3.2 Results for Headlines.txt

```
{'num_total_words': 4543291, 'num_unique_words': 191435,
'num_of_paragraphs': 500000}
Number of words in only 1 document: 99277
Percentage of words in only 1 document: 0.5185937785671376

Top 100
          to: Collection frequency: 118900, Document frequency:109858
          in: Collection frequency:  88346, Document frequency:83810
         the: Collection frequency:  84410, Document frequency:72954
          of: Collection frequency:  76126, Document frequency:70057
         for: Collection frequency:  67594, Document frequency:65772
         and: Collection frequency:  55417, Document frequency:51216
          on: Collection frequency:  41834, Document frequency:40816
           a: Collection frequency:  37843, Document frequency:35023
        with: Collection frequency:  31812, Document frequency:31320
          at: Collection frequency:  31605, Document frequency:31108
         new: Collection frequency:  26807, Document frequency:26298
        2015: Collection frequency:  20878, Document frequency:20530
          is: Collection frequency:  17944, Document frequency:17539
          by: Collection frequency:  17338, Document frequency:16765
          as: Collection frequency:  16658, Document frequency:16054
        from: Collection frequency:  16159, Document frequency:16020
       after: Collection frequency:  12967, Document frequency:12922
      market: Collection frequency:  10890, Document frequency: 9746
        over: Collection frequency:   9838, Document frequency: 9795
          be: Collection frequency:   9274, Document frequency: 9178
          up: Collection frequency:   9080, Document frequency: 8980
   announces: Collection frequency:   8381, Document frequency: 8376
      global: Collection frequency:   8104, Document frequency: 8013
        says: Collection frequency:   7881, Document frequency: 7867
        more: Collection frequency:   7644, Document frequency: 7542
         man: Collection frequency:   7601, Document frequency: 7530
         you: Collection frequency:   7512, Document frequency: 7034
        will: Collection frequency:   7510, Document frequency: 7448
        your: Collection frequency:   7367, Document frequency: 7068
```

```
      out: Collection frequency:   7312, Document frequency: 7263
september: Collection frequency:   7173, Document frequency: 7121
    first: Collection frequency:   7105, Document frequency: 7056
       us: Collection frequency:   7074, Document frequency: 7012
      how: Collection frequency:   6726, Document frequency: 6683
      are: Collection frequency:   6687, Document frequency: 6577
     this: Collection frequency:   6485, Document frequency: 6429
   police: Collection frequency:   6477, Document frequency: 6427
      day: Collection frequency:   6388, Document frequency: 6304
    world: Collection frequency:   6290, Document frequency: 6202
      not: Collection frequency:   6032, Document frequency: 5979
   report: Collection frequency:   5968, Document frequency: 5897
    about: Collection frequency:   5730, Document frequency: 5675
       it: Collection frequency:   5634, Document frequency: 5455
     week: Collection frequency:   5399, Document frequency: 5319
    video: Collection frequency:   5122, Document frequency: 5065
      its: Collection frequency:   5100, Document frequency: 5006
       no: Collection frequency:   4974, Document frequency: 4761
      has: Collection frequency:   4950, Document frequency: 4913
       an: Collection frequency:   4936, Document frequency: 4876
     that: Collection frequency:   4835, Document frequency: 4783
      top: Collection frequency:   4811, Document frequency: 4760
      inc: Collection frequency:   4810, Document frequency: 4283
     into: Collection frequency:   4770, Document frequency: 4759
      one: Collection frequency:   4713, Document frequency: 4570
  against: Collection frequency:   4662, Document frequency: 4647
      off: Collection frequency:   4606, Document frequency: 4544
     home: Collection frequency:   4562, Document frequency: 4476
      his: Collection frequency:   4523, Document frequency: 4391
   school: Collection frequency:   4490, Document frequency: 4388
        2: Collection frequency:   4488, Document frequency: 4411
     back: Collection frequency:   4417, Document frequency: 4382
      u.s: Collection frequency:   4398, Document frequency: 4369
 business: Collection frequency:   4398, Document frequency: 4296
    china: Collection frequency:   4390, Document frequency: 4308
    group: Collection frequency:   4378, Document frequency: 4297
    state: Collection frequency:   4301, Document frequency: 4208
     what: Collection frequency:   4284, Document frequency: 4229
      can: Collection frequency:   4273, Document frequency: 4247
 research: Collection frequency:   4243, Document frequency: 4138
     have: Collection frequency:   4211, Document frequency: 4166
      win: Collection frequency:   4168, Document frequency: 4146
     best: Collection frequency:   4159, Document frequency: 4090
      two: Collection frequency:   4126, Document frequency: 4070
      who: Collection frequency:   4126, Document frequency: 4078
   review: Collection frequency:   4123, Document frequency: 4107
```

```
        open: Collection frequency:   4104, Document frequency: 4076
         get: Collection frequency:   4075, Document frequency: 4053
         now: Collection frequency:   4048, Document frequency: 4032
        city: Collection frequency:   4022, Document frequency: 3935
    industry: Collection frequency:   4001, Document frequency: 3964
           : Collection frequency:   3827, Document frequency: 3591
     million: Collection frequency:   3775, Document frequency: 3721
         all: Collection frequency:   3770, Document frequency: 3725
        time: Collection frequency:   3736, Document frequency: 3677
         big: Collection frequency:   3725, Document frequency: 3647
        year: Collection frequency:   3689, Document frequency: 3631
    launches: Collection frequency:   3664, Document frequency: 3662
           3: Collection frequency:   3635, Document frequency: 3598
        2016: Collection frequency:   3589, Document frequency: 3562
         why: Collection frequency:   3584, Document frequency: 3561
         set: Collection frequency:   3564, Document frequency: 3552
        down: Collection frequency:   3551, Document frequency: 3531
        show: Collection frequency:   3540, Document frequency: 3499
           i: Collection frequency:   3489, Document frequency: 3200
           5: Collection frequency:   3486, Document frequency: 3438
        news: Collection frequency:   3472, Document frequency: 3413
        help: Collection frequency:   3460, Document frequency: 3431
         cup: Collection frequency:   3448, Document frequency: 3396
          10: Collection frequency:   3407, Document frequency: 3374
       their: Collection frequency:   3406, Document frequency: 3315
Top 500th word
      russia: Collection frequency:   1180, Document frequency: 1175
Top 1000th word
    district: Collection frequency:    674, Document frequency:  666
Top 5000th word
     orioles: Collection frequency:    120, Document frequency:  118
```