3. Indexing

August 12, 2018

Saumay Agrawal
16BCE1151

In [1]:
```
# Importing necessary modules
from prettytable import PrettyTable
import requests
from bs4 import BeautifulSoup
```

In [2]:
```
# Preprocessing a list of words of a document
def preprocess(words):
    for i in range(len(words)):
        word = words[i].strip()
        if not word[0].isalnum():
            word = word[1:]
        if not word[-1].isalnum():
            word = word[:-1]
        words[i] = word.lower()
    return words

# Get the offsets of a word in a list of words
def getoffsets(words, word):
    offsets = []
    for i in range(len(words)):
        if words[i]==word:
            offsets.append(i)
    return offsets

# Create index of passed document 'doc' based on its 'doc_id'
def createindex(doc, doc_id):
    words = doc.split(' ')
    words = [word for word in words if word!='']
    words = preprocess(words)
    index = {}
    for i in set(words):
        offsets = getoffsets(words, i)
        postings = [doc_id, offsets]
        index[i] = [words.count(i), [postings]]
```

```
            return index

        # Merge the doc_index and index
        def appendindex(doc_index, index):
            for key in doc_index.keys():
                if key in index.keys():
                    index[key][0] += doc_index[key][0]
                    index[key][1] += doc_index[key][1]
                else:
                    index[key] = doc_index[key]
            return index

        # Print the index in tabular format
        def printindex(index):
            table = PrettyTable(['Word', 'Frequency', 'Postings'])
            for word in sorted(index.keys()):
                frequency, postings = index[word][0], index[word][1]
                table.add_row([word, frequency, postings])
            print(table)
```

In [3]:
```
        # Create the index of sample input files
        index = {}
        files = ['input1.txt', 'input2.txt', 'input3.txt']
        doc_id = 0
        for file in files:
            doc_id += 1
            content = None
            with open(file, 'r') as content_file:
                content = content_file.read()
            doc_index = createindex(content, doc_id)
            index = appendindex(doc_index, index)
```

In [4]:
```
printindex(index)
```

```
+--------------+-----------+---------------------------------+
|     Word     | Frequency |             Postings            |
+--------------+-----------+---------------------------------+
|     and      |     1     |            [[2, [8]]]           |
|    black     |     3     | [[1, [3]], [2, [11]], [3, [0]]] |
|     cash     |     1     |            [[2, [1]]]           |
|  corruption  |     1     |            [[1, [0]]]           |
|   creation   |     1     |            [[2, [9]]]           |
|   currency   |     1     |            [[3, [8]]]           |
| denomination |     1     |            [[3, [7]]]           |
|  excessive   |     1     |            [[2, [0]]]           |
|  generation  |     1     |            [[1, [5]]]           |
|     high     |     1     |            [[3, [6]]]           |
|      in      |     3     |  [[1, [2]], [2, [4]], [3, [5]]] |
```

```
|      is      |     1     |            [[3, [2]]]            |
|    money     |     3     | [[1, [4]], [2, [12]], [3, [1]]] |
| non-payment  |     1     |            [[2, [5]]]            |
|      of      |     2     |          [[2, [6, 10]]]          |
|    result    |     1     |            [[2, [3]]]            |
|   results    |     1     |            [[1, [1]]]            |
|    stored    |     1     |            [[3, [4]]]            |
|     tax      |     1     |            [[2, [7]]]            |
| transactions |     1     |            [[2, [2]]]            |
|   usually    |     1     |            [[3, [3]]]            |
+--------------+-----------+----------------------------------+
```

```
In [5]: # Create the index of four of Robert Frost's poems
        index = {}
        links = ['https://www.poemhunter.com/poem/the-road-not-taken/', 'https://www.poemhunter
```

```
In [6]: for id in range(len(links)):
            link = links[id]
            response = requests.get(link)
            soup = BeautifulSoup(response.content, 'html.parser')
            poem = str(soup.find_all('p')[1])
            poem = poem[4:-4].replace('<br/>', ' ').strip()
            print('Parsed poem #{}:\n{}\n'.format(id, poem))
            poem_index = createindex(poem, id)
            index = appendindex(poem_index, index)
```

```
Parsed poem #0:
Two roads diverged in a yellow wood, And sorry I could not travel both And be one traveler, lo

Parsed poem #1:
Whose woods these are I think I know. His house is in the village, though;  He will not see me

Parsed poem #2:
Nature's first green is gold, Her hardest hue to hold. Her early leaf's a flower;  But only so

Parsed poem #3:
A voice said, Look me in the stars And tell me truly, men of earth, If all the soul-and-body s
```

```
In [7]: for word in sorted(index.keys()):
            freq, postings = index[word][0], index[word][1]
            print('{}: {}'.format(word, freq))
            for i in postings:
                print('\t{}'.format(i))
            print()
```

```
a: 7
        [0, [4, 115, 126]]
```

```
        [1, [40, 60]]
        [2, [13]]
        [3, [0]]

about: 1
        [0, [69]]

ages: 2
        [0, [118, 120]]

all: 2
        [0, [141]]
        [3, [16]]

an: 1
        [2, [18]]

and: 15
        [0, [7, 14, 21, 45, 55, 72, 119, 128, 137]]
        [1, [46, 78, 86, 94, 101]]
        [3, [8]]

another: 1
        [0, [91]]

are: 2
        [1, [3, 83]]

as: 5
        [0, [25, 27, 41, 43, 59]]

ask: 1
        [1, [63]]

back: 1
        [0, [108]]

be: 2
        [0, [15, 111]]

because: 1
        [0, [51]]

before: 2
        [1, [98, 105]]

bells: 1
        [1, [59]]
```

```
bent: 1
        [0, [33]]

better: 1
        [0, [49]]

between: 1
        [1, [43]]

birth: 1
        [3, [27]]

black: 1
        [0, [84]]

both: 2
        [0, [13, 73]]

but: 2
        [1, [88]]
        [2, [15]]

by: 1
        [0, [136]]

can: 1
        [2, [38]]

claim: 1
        [0, [50]]

come: 1
        [0, [107]]

could: 2
        [0, [10, 29]]

dark: 1
        [1, [85]]

darkest: 1
        [1, [50]]

dawn: 1
        [2, [31]]

day: 2
```

```
        [0, [92]]
        [2, [35]]

deep: 1
        [1, [87]]

difference: 1
        [0, [143]]

diverged: 2
        [0, [2, 124]]

doubted: 1
        [0, [102]]

down: 2
        [0, [23]]
        [2, [33]]

downy: 1
        [1, [79]]

early: 1
        [2, [11]]

earth: 1
        [3, [14]]

easy: 1
        [1, [76]]

eden: 1
        [2, [26]]

equally: 1
        [0, [76]]

evening: 1
        [1, [51]]

ever: 1
        [0, [106]]

fair: 1
        [0, [44]]

far: 1
        [0, [26]]
```

```
farmhouse: 1
        [1, [41]]

fill: 1
        [1, [26]]

first: 2
        [0, [89]]
        [2, [1]]

flake: 1
        [1, [80]]

flower: 1
        [2, [14]]

for: 3
        [0, [60, 90]]
        [3, [26]]

frozen: 1
        [1, [47]]

gives: 1
        [1, [56]]

go: 2
        [1, [97, 104]]

goes: 1
        [2, [32]]

gold: 2
        [2, [4, 37]]

grassy: 1
        [0, [54]]

green: 1
        [2, [2]]

grief: 1
        [2, [29]]

had: 2
        [0, [65, 82]]
```

```
hardest: 1
        [2, [6]]

harness: 1
        [1, [58]]

has: 1
        [0, [139]]

have: 1
        [1, [90]]

having: 1
        [0, [46]]

he: 2
        [1, [15, 55]]

hence: 1
        [0, [121]]

her: 2
        [2, [5, 10]]

here: 1
        [1, [21]]

his: 3
        [1, [8, 24, 57]]

hold: 1
        [2, [9]]

horse: 1
        [1, [32]]

hour: 1
        [2, [19]]

house: 1
        [1, [9]]

how: 1
        [0, [95]]

hue: 1
        [2, [7]]
```

```
i: 14
        [0, [9, 19, 28, 86, 101, 104, 109, 129, 130]]
        [1, [4, 6, 89, 99, 106]]

if: 3
        [0, [103]]
        [1, [64]]
        [3, [15]]

in: 6
        [0, [3, 34, 78, 125]]
        [1, [11]]
        [3, [5]]

is: 3
        [1, [10, 66]]
        [2, [3]]

it: 3
        [0, [32, 52]]
        [1, [35]]

just: 1
        [0, [42]]

keep: 1
        [1, [93]]

kept: 1
        [0, [87]]

know: 1
        [1, [7]]

knowing: 1
        [0, [94]]

lake: 1
        [1, [48]]

lay: 1
        [0, [77]]

leads: 1
        [0, [97]]

leaf: 2
        [2, [21, 24]]
```

```
leaf's: 1
        [2, [12]]

leaves: 1
        [0, [79]]

less: 1
        [0, [134]]

little: 1
        [1, [31]]

long: 1
        [0, [18]]

look: 1
        [3, [3]]

looked: 1
        [0, [22]]

lovely: 1
        [1, [84]]

made: 1
        [0, [140]]

me: 3
        [1, [19]]
        [3, [4, 10]]

men: 1
        [3, [12]]

miles: 2
        [1, [95, 102]]

mistake: 1
        [1, [68]]

morning: 1
        [0, [75]]

much: 1
        [3, [23]]

must: 1
```

```
        [1, [33]]

my: 1
        [1, [30]]

nature's: 1
        [2, [0]]

near: 1
        [1, [42]]

no: 1
        [0, [80]]

not: 3
        [0, [11]]
        [1, [17]]
        [3, [21]]

nothing: 1
        [2, [36]]

of: 3
        [1, [52, 75]]
        [3, [13]]

oh: 1
        [0, [85]]

on: 1
        [0, [98]]

one: 3
        [0, [16, 24, 133]]

only: 2
        [1, [70]]
        [2, [16]]

other: 2
        [0, [40]]
        [1, [71]]

passing: 1
        [0, [63]]

pay: 1
        [3, [25]]
```

```
perhaps: 1
        [0, [47]]

promises: 1
        [1, [91]]

queer: 1
        [1, [36]]

really: 1
        [0, [68]]

roads: 2
        [0, [1, 123]]

said: 1
        [3, [2]]

same: 1
        [0, [71]]

sank: 1
        [2, [27]]

scars: 1
        [3, [19]]

see: 1
        [1, [18]]

shake: 1
        [1, [61]]

shall: 1
        [0, [110]]

should: 1
        [0, [105]]

sigh: 1
        [0, [116]]

sleep: 2
        [1, [100, 107]]

snow: 1
        [1, [29]]
```

```
so: 3
        [2, [17, 25, 30]]

some: 1
        [1, [67]]

somewhere: 1
        [0, [117]]

sorry: 1
        [0, [8]]

soul-and-body: 1
        [3, [18]]

sound's: 1
        [1, [72]]

stars: 1
        [3, [7]]

stay: 1
        [2, [39]]

step: 1
        [0, [81]]

stood: 1
        [0, [20]]

stop: 1
        [1, [38]]

stopping: 1
        [1, [20]]

subsides: 1
        [2, [22]]

sweep: 1
        [1, [74]]

tell: 1
        [3, [9]]

telling: 1
        [0, [112]]
```

```
that: 3
        [0, [61, 74, 138]]

the: 17
        [0, [35, 39, 48, 62, 70, 88, 132, 142]]
        [1, [12, 44, 49, 53, 69, 73, 81]]
        [3, [6, 17]]

them: 1
        [0, [67]]

then: 2
        [0, [37]]
        [2, [20]]

there: 2
        [0, [64]]
        [1, [65]]

these: 1
        [1, [2]]

think: 2
        [1, [5, 34]]

this: 1
        [0, [113]]

though: 2
        [0, [58]]
        [1, [14]]

to: 13
        [0, [30, 99]]
        [1, [22, 37, 62, 92, 96, 103]]
        [2, [8, 23, 28, 34]]
        [3, [24]]

too: 1
        [3, [22]]

took: 2
        [0, [38, 131]]

travel: 1
        [0, [12]]
```

```
traveled: 1
        [0, [135]]

traveler: 1
        [0, [17]]

trodden: 1
        [0, [83]]

truly: 1
        [3, [11]]

two: 2
        [0, [0, 122]]

undergrowth: 1
        [0, [36]]

up: 1
        [1, [27]]

village: 1
        [1, [13]]

voice: 1
        [3, [1]]

wanted: 1
        [0, [56]]

was: 1
        [0, [53]]

watch: 1
        [1, [23]]

way: 2
        [0, [96, 100]]

wear: 1
        [0, [57]]

were: 1
        [3, [20]]

where: 1
        [0, [31]]
```

```
whose: 1
        [1, [0]]

will: 1
        [1, [16]]

wind: 1
        [1, [77]]

with: 2
        [0, [114]]
        [1, [28]]

without: 1
        [1, [39]]

wood: 2
        [0, [6, 127]]

woods: 4
        [1, [1, 25, 45, 82]]

worn: 1
        [0, [66]]

year: 1
        [1, [54]]

yellow: 1
        [0, [5]]

yet: 1
        [0, [93]]
```