Assignment 2

Fall 2016
CS834 Introduction to Information Retrieval
Dr. Michael Nelson

Mathew Chaney

October 11, 2016

Contents

List of Figures

Listings

List of Tables

1 Question 4.1

1.1 Question

Plot rank-frequency curves (using a log-log graph) for words and bigrams in the Wikipedia collection available through the book website (http://www.search-engines-book.com). Plot a curve for the combination of the two. What are the best values for the parameter c for each curve?

1.2 Resources

The textbook *Search Engines: Information Retrieval in Practice* [1], the Python programming language [2], the R programming language [?] and the BeautifulSoup python library [3] were used to answer this question.

1.3 Answer

The wc.py script 1 was used to locate each file of the Wikipedia collection obtained from the book download page, available at http://www.search-engines-book.com. The BeautifulSoup library was used to strip out the HTML tags and then the nltk library [?] was used to tokenize the text. The individual words were counted manually and the nltk library [?] was used to count the bigrams.

The word count graph can be found in Figure 1 and the bigram count graph can be found in Figure 2.

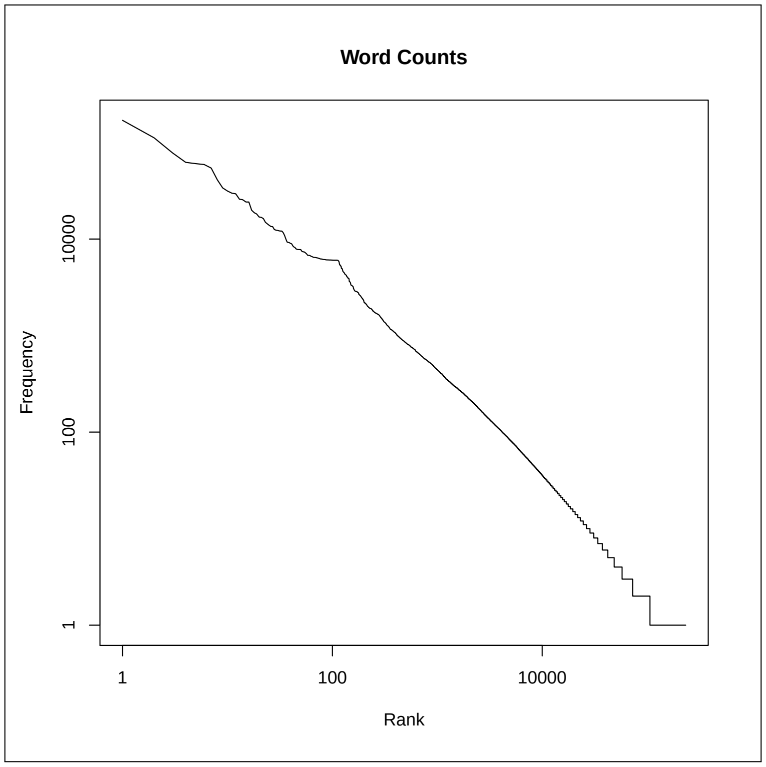

Figure 1: Word Counts for Small Wikipedia Corpus

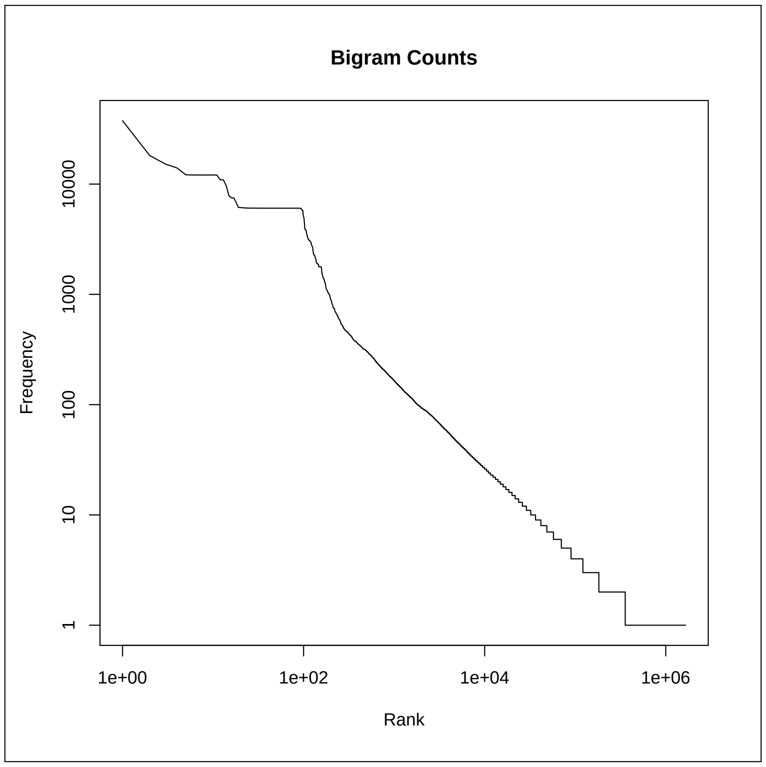

Figure 2: Bigram Counts for Small Wikipedia Corpus

2 Appendix A

```python
#!/usr/bin/python

import argparse
import os
import operator
import sys
import nltk
from os.path import isdir, isfile
from bs4 import BeautifulSoup

class WordCounter(object):
    def __init__(self, root):
        self.tokenizer = nltk.RegexpTokenizer(r'\w+')
        self.root = root
        self.wmap = {}
        self.bgmap = {}
        self.filelist = []
        self.visited = 0

    def getfiles(self, folder=''):
        items = os.listdir(self.root + folder)
        for item in items:
            filepath = self.root + folder + os.sep + item
            if isfile(filepath):
                self.filelist.append(filepath)
            elif isdir(filepath):
                self.getfiles(folder + os.sep + item)

    def count(self, filepath):
        sys.stdout.write("\rprocessing document #%i" % self.visited)
        sys.stdout.flush()
        with open(filepath) as infile:
            text = BeautifulSoup(infile.read(), 'html.parser').get_text()
            tokens = self.tokenizer.tokenize(text)
        for s in tokens:
            if not self.wmap.has_key(s):
                self.wmap[s] = 0
            self.wmap[s] = self.wmap[s] + 1
        for b in nltk.bigrams(tokens):
            if not self.bgmap.has_key(b):
                self.bgmap[b] = 0
            self.bgmap[b] = self.bgmap[b] + 1
        self.visited = self.visited + 1

    def writeresults(self):
        with open('wordcount', 'w') as outfile:
            for k, v in sorted(self.wmap.items(), key=operator.itemgetter(1), reverse=True):
                outfile.write(str(v) + '\t' + k.encode('utf-8') + '\n')
        with open('bigramcount', 'w') as outfile:
            for k, v in sorted(self.bgmap.items(), key=operator.itemgetter(1), reverse=True):
                outfile.write(str(v) + '\t' + k[0].encode('utf-8') + '\t' + k[1].encode('utf-8') + '\n')

    def run(self):
        print 'delving into "{0}"'.format(self.root)
        self.getfiles()
        print 'found {0} documents'.format(len(self.filelist))
        map(self.count, self.filelist)
        print '\nfound {0} words'.format(len(self.wmap))
        print 'found {0} bigrams'.format(len(self.bgmap))
        self.writeresults()

if __name__ == '__main__':
    parser = argparse.ArgumentParser('word count')
    parser.add_argument('-root', '-r', help='the root directory for parsing', default='en')
    args = parser.parse_args()
    wc = WordCounter(args.root)
    wc.run()
```

Listing 1: wc.py

```
#! /usr/bin/Rscript

plotgraph <- function(infile, outfile, title) {
    data <- read.table(infile)
    x <- seq(1, length(data$V1))
    y <- data$V1

    pdf(outfile)
    plot(x, y, type='l', log='xy', main=title,
        ylab='Frequency', xlab='Rank')
    dev.off()
}

plotgraph('wordcount', 'wc.pdf', 'Word Counts')
plotgraph('bigramcount', 'bg.pdf', 'Bigram Counts')
```

Listing 2: buildgraphs.R

3 References

[1] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Pearson, first edition, February 2009.

[2] The Python Programming Language. Available at: `https://www.python.org/`. Accessed: 2016/09/17.

[3] Leonard Richardson. Beautiful Soup. Available at: `https://www.crummy.com/software/beautifulsoup/`. Accessed: 2016/09/20.