# CS 751: Assignment 3

Bharath Kongara

Spring 2015

# Contents

# 1 Question 1

- For the text you saved for the 10000 URIs from A1, Q2:
  - Use the boilerpipe software to remove the HTML templates from all HTML pages (document how many pages link from the tweets were non-HTML and had to be skipped)
  - https://code.google.com/p/boilerpipe/
  - WSDM 2010 paper: http://www.l3s.de/ kohlschuetter/boilerplate/
- For how many of the 10000 URIs was boilerpipe successful?
  - Compare the total words, unique words, and byte sizes before and after use of boilerpipe.
- For what classes of pages was it successful?
- For what classes of pages was it unsuccessful?
- Provide examples of both successful and unsuccessful removals and discuss at length.

## 1.1 Solution

- Installed jusText[1] to remove templates from HTML files of 10000 URIs.
- jusText is a tool for removing boilerplate content such as navigation links, headers and footers from HTML pages.

## 1.2 Code Listing

Below is the python code that is used to remove HTML templates from all HTML pages of 10000 URIs.

```
1  import urllib2
2  import json
3  import justext
4  f = open('statusoutputs10','r+')
5  i=9000
6  for line in f:
7      try:
8          data = json.loads(line)
9          page = urllib2.urlopen(data['finalurl']).read()
10         paragraphs = justext.justext(page, justext.get_stoplist('English'))
11
12         i=i+1
13         for paragraph in paragraphs:
14             if not paragraph.is_boilerplate:
15                 filename = 'textfromhtml'+str(i)
16                 outputfile = open(filename, 'w')
17                 outputfile.write(paragraph.text.encode('utf-8') + '\n')
18     except Exception as e:
19         print data['finalurl']
20         continue
```

Listing 1: Python program to remove HTML templates

Here is the python code that is used to count total words, unique words and frequency. Basic parsing of the text files is referred from wordcount python library[2].

```
1
2  import sys
3  def sort_by_value(item):
4      return item[-1]
5
6  def build_dict(filename):
7      f = open(filename, 'rU')
8      words = f.read().split()
9      count = {}
10
11     for word in words:
12         word = word.lower()
```

```
13          if word not in count:
14              count[word] = 1
15          else:
16              count[word] += 1
17
18      f.close()
19      return count
20
21  def print_words(filename):
22      dict = build_dict(filename)
23
24      for word in sorted(dict.keys()):
25          print word, dict[word]
26
27  def print_top(filename):
28      count = build_dict(filename)
29      i = 0
30
31      items = sorted(count.items(), key=sort_by_value, reverse=True)
32      for item in items[:20]:
33          print item[0] + ': ' + str(item[1]) + ' times'
34          i += 1
35
36  def main():
37      if len(sys.argv) != 3:
38          print 'usage: ./wordcount.py {--count | --topcount} file'
39          sys.exit(1)
40
41      option = sys.argv[1]
42      filename = sys.argv[2]
43      if option == '--count':
44        print_words(filename)
45      elif option == '--topcount':
46        print_top(filename)
47      else:
48        print 'unknown option: ' + option
49      sys.exit(1)
50
51  if __name__ == '__main__':
52      main()
```

Listing 2: Python program for finding unique words and total words

Table 1: Original and After jusText

| | Before jusText | After jusText |
|---|---|---|
| Total Bytes | 813,340,810 | 650,600 |
| Total Words | 305,559 and | 105,559 |
| Unique Words | 40,011 and | 20,611 |

### 1.3 Comparing total words, unique words and byte sizes

- Comparison of total words, unique words and total bytes before use of jusText is shown in table 1.
- Out of my 10000 URIs, 6200 were unique URIs. In this list of URIs 4643 was jusText successful.
- Remaining 1566 were skipped due to errors like '404', could not resolve host, etc. as shown below.

```
$ curl -I http://waniko.info/u70xbs/
curl: (6) Could not resolve host: waniko.info

$ curl -I http://truffle.upper.jp/44p3z
HTTP/1.1 302 Found
Date: Sat, 04 Apr 2015 04:23:10 GMT
Server: Apache/2.0.58 (Unix) mod_bwshare/0.2.0 PHP/5.3.8
Location: http://err.lolipop.jp/404.html
```

- Few of them are non-HTML and they were skipped due to errors as shown below.

```
8 URIs pointing to Images of Instagram
3 URIS from facebook site
2 URIs which are pointing to sites with no content.
```

- Examples of URIs for which jusText was successful are listed below. Pages which are properly structured with HTML 5 tags like section, caption, footer, header are successful.
  - http://appleinsider.com/articles/15/02/09/rumor-apple-to-again-stay-out-of-megapixel-race
  - http://www.aztecanoticias.com.mx/notas/finanzas/212592/bolsas-de-europa-en

# Rumor: Apple to again stay out of megapixel race with 8MP camera in 'iPhone 6s'

By Katie Marsal
Monday, February 09, 2015, 08:15 am PT (11:15 am ET)

Apple has historically downplayed the significance of megapixels in measuring image quality, and the company may stick to that approach once again later this year with another 8-megapixel camera in its next-generation iPhone, according to a new rumor.

Alemania impuso récord de... | Más Notas | Dólar se cotiza en 14.70 pesos a...

# Bolsas de Europa, en incertidumbre por Grecia

Plazas bursátiles concluyen a la baja ante desplome de mercado griego; Madrid reporta la mayor pérdida, de 1.97%.

Fuente Notimex
09 de febrero de 2015
14:02 hrs



Bolsa alemana cede 1.69% (Imagen: Reuters).



Madrid, España.-La mayoría de las principales **bolsas** de valores de **Europa** concluyeron hoy operaciones a la baja, arrastrada por el desplome del mercado griego, ante la incertidumbre que prevalece sobre su futuro financiero.

Con excepción del mercado de Zúrich, que terminó con un avance de 0.55 por ciento, el resto de las **bolsa** cerró este lunes con números rojos, resaltando el índice Ibex-35 de Madrid, que cerró con un retroceso de 1.97 por ciento.

En el mercado de divisas, el Banco Central Europeo (BCE) colocó este lunes la cotización oficial del euro frente al dólar en 1.12, una baja de 1.50 por ciento respecto a la jornada anterior.

La onza de oro en el World Gold Council se cotizó en 1,242.10 dólares a la venta y

www.aztecanoticias.com.mx/notas/finanzas/217318/bolsa-de-nueva-york-se-recupera-por-datos-de-empra, un incremento en relación a la jornada anterior, cuando

**MÁS NOTICIAS**
de Finanzas


Bolsa de Nueva York se recupera por datos de empleo


INEGI: Inversión en industria creció en enero


Exportaciones mexicanas a Canadá crecen


Bolsas de valores de Europa, con ligeras ganancias


Mercado laboral mejora en Estados Unidos

Ver más notas

**VIDEOS RELACIONADOS**
de Finanzas


Video: Escenario para la Pasión de Cristo en Iztapalapa, listo


Video: Finanzas y Negocios, 1 abril 2015


Video: Canacope pide a Profeco verificar precios

- Classes of pages for which jusText was not successful are given below.
  - http://news.uvs.jp/1D4
  - http://gharedly.com/
- Pages which are cluttered with HTML elements and have not followed guidelines for HTML page design are not successful.

غردلى بالخير


كل الأشياء ترحل ولاتعود

إلا الدعاء : يرحل بالرجاء..ويعود بالعطاء

تسجيل الدخول

ماهي خدمة غردلى ؟

ماهي اهم مميزات الخدمة

ماهي خدمة ضيفني و اضيفك ؟

سياسة الخصوصية الشروط و الاحكام

Gharedly.Com

短縮URLサービス

## 「チェインクロニクル」，リングガチャに新キャラ・魔法兵団隊長エ...

http://www.4gamer.net/games/223/G022384/20141112006/

上記URLへ移動します。
よろしいですか?

2015/04/04 今年10回目 中国船が領海侵入
2015/04/04 盗撮容疑の芸人引退 吉本発表
2015/04/04 脱出完了まで6時間 青函発煙
2015/04/04 Apple Watchの予約時間が判明！ 日本時間4月10日、午後4時1分から
2015/04/04 ようじ1万本でマリオOP画面
2015/04/04 つんくが声帯摘出公表 声失う
2015/04/04 月の影の中を飛ぶ、暗く少し恐ろしい光景
2015/04/04 4年64億円MLB右腕、薬物陽性
2015/04/04 うわぁ…と思わず声が出そう。チームラボの花と一体化するアート
2015/04/04 ＩＴ技術者不足でアジアの学生を日本に　NHKニュース
2015/04/04 男子ゴコロをくすぐる「STI Performance Concept」
2015/04/04 自撮り棒ならぬ「自撮り鞄」という提案 : ギズモード・ジャパン
2015/04/04 自撮り棒ならぬ「自撮り鞄」という提案
2015/04/04 週刊「しょうもないWebアプリをつくる」創刊号 – アクセスカウンター | CreativeStyle
2015/04/04 フリーハンドで有名ロゴを一発描き、もはや芸術か！
2015/04/04 首都圏の私大下宿生、１日生活費が初めて９００円割る：朝日新聞デジタル
2015/04/04 Chromeが重いな～と思ったら32bit版を使ってた件。64bitにしたら快適でワロタｗｗｗ : IT速報
2015/04/04 できなければ死んだも同じ…中高生のインフラ「LINE」の実態 - CNET Japan
2015/04/04 目撃情報を元にしたオーロラのグローバルマップで夢が叶うかも

運営元 | お問い合わせ

# 2 Question 2

- Collection 1: Extract all the unique terms and their frequency from the 10000 files
- Collection 2: Extract all the unique terms and their frequency of the 10000 files after running boilerpipe.
- Construct a table with the top 50 terms from each collection. - Find a common stop word list. How many of the 50 terms are on that stop word list?
- For both collections, construct a graph with the x-axis as word rank,and y-axis as word frequency. - Do either follow a Zipf distribution? Support your answer.

## 2.1 Solution

Each line of text file are parsed by splitting words on space and constructed dictionary to get the unique word list. Here is the Python program which calculates top 50 unique word frequencies.

```
1  import os, sys
2  page = ''
3  wordFrequencyDict = {}
4  try:
5          inputFile = open('total.txt', 'r')
6          page = inputFile.read()
7          inputFile.close()
8  except:
9          exc_type, exc_obj, exc_tb = sys.exc_info()
10         fname = os.path.split(exc_tb.tb_frame.f_code.co_filename)[1]
11         print(fname, exc_tb.tb_lineno, sys.exc_info() )
12
13 tokens = page.split(' ')
14 for t in tokens:
15         wordFrequencyDict.setdefault(t, 0)
16         wordFrequencyDict[t] = wordFrequencyDict[t] + 1
17 wordFrequencyDict = sorted(wordFrequencyDict.items(), key=lambda x:x[1], reverse=True)
18
19 totalWords = 0
20 for tup in wordFrequencyDict:
21         totalWords = tup[1] + totalWords
22
23 print 'totalWords:', totalWords
24 print 'totalUniqueWords:', len(wordFrequencyDict)
```

Listing 3: Python program for calculating unique words frequency

## 2.2 Find terms which are common in stop words list:

- 16 terms from Table 2 are common with the stop words list that I retrieved from `http://www.ranks.nl/stopwords`.
- 38 terms from Table 3 are common with the stop words list.
- Stop words are listed in the file stop-words.txt.
- Both collections before and after running boilerpipe are plotted on a single graph. Below is the R code which is used to plot the data.

```
1
2
3
4  htmlF <- read.table('graphhtml.txt', header=T)
5  htmlData <- rep(htmlF[,1])
6
7  textF<- read.table('graphtext.txt', header=T)
8  textData <- rep(textF[,1])
9  xlimit <- c(0,50)
10 ylimit <- c(400,4000)
11
12 plot(htmlData, type='l', col='blue',xlim=xlimit,ylim=ylimit, xlab='Word Rank', ylab='Word
       Frequency', main='Distribution of terms before boilerplate \nremoval (blue) and after
       boilerplate removal (red)')
13
14 lines(textData, type='o', col='red')
```

Listing 4: R program to plot both collections

- As shown in the figure below, both plots follow Zipfian Distribution[3].
- The frequency of each word is close to inversely proportional to its rank in the frequency table. This is known as the power law.
- As shown in the figure below, most frequent word occurs approximately twice as the second most frequency word.

Table 2: Rank, term and frequency from jusText file

| RANK | TERM | FREQUENCY |
|---|---|---|
| 1 | the | 1164 |
| 2 | and | 925 |
| 3 | to | 809 |
| 4 | of | 899 |
| 5 | a | 771 |
| 6 | you | 625 |
| 7 | I | 500 |
| 8 | is | 423 |
| 9 | that | 304 |
| 10 | you | 289 |
| 11 | for | 280 |
| 12 | it | 253 |
| 13 | with | 249 |
| 14 | or | 245 |
| 15 | on | 229 |
| 16 | this | 222 |
| 17 | we | 221 |
| 18 | s | 218 |
| 19 | your | 201 |
| 20 | as | 176 |
| 21 | from | 170 |
| 22 | domain | 150 |
| 23 | will | 146 |
| 24 | are | 140 |
| 25 | not | 137 |
| 26 | by | 131 |
| 27 | be | 120 |
| 28 | at | 117 |
| 29 | have | 114 |
| 30 | all | 110 |
| 31 | can | 108 |
| 32 | my | 108 |
| 33 | t | 99 |
| 34 | my | 98 |
| 35 | about | 94 |
| 36 | when | 92 |
| 37 | like | 86 |
| 38 | use | 85 |
| 39 | so | 84 |
| 40 | more | 83 |
| 41 | but | 80 |
| 42 | do | 79 |
| 43 | any | 79 |
| 44 | about | 77 |
| 45 | her | 77 |
| 46 | just | 76 |
| 47 | one | 70 |
| 48 | who | 68 |
| 49 | has | 67 |
| 50 | if | 67 |

Table 3: Top 50 terms extracted from HTML files

| RANK | TERM | FREQUENCY |
|---|---|---|
| 1 | div | 5921 |
| 2 | = | 3933 |
| 3 | the | 3711 |
| 4 | ¡a | 3511 |
| 5 | to | 3408 |
| 6 | and | 3390 |
| 7 | a | 3270 |
| 8 | of | 3110 |
| 9 | { | 3032 |
| 10 | <li ><a | 2913 |
| 11 | <span | 2878 |
| 12 |  | 2773 |
| 13 | in | 2683 |
| 14 | {. | 2583 |
| 15 | for | 2480 |
| 16 | - | 2310 |
| 17 | <li | 2237 |
| 18 | at | 2137 |
| 19 | var | 2031 |
| 20 | point:false, | 1997 |
| 21 | on | 1896 |
| 22 | + | 1793 |
| 23 |  | 1633 |
| 24 | is | 1489 |
| 25 | target='self' | 1211 |
| 26 | ? | 1031 |
| 27 | position:'left'}, | 937 |
| 28 | yous | 820 |
| 29 | your | 708 |
| 30 | </li > | 681 |
| 31 | class="user-in" | 608 |
| 32 | with | 585 |
| 33 | I | 565 |
| 34 | user | 550 |
| 35 | target=" | 520 |
| 36 | & | 482 |
| 37 | 2014 | 472 |
| 38 | </a > | 460 |
| 39 | December | 452 |
| 40 | 0 | 400 |
| 41 | rel='fll' | 368 |
| 42 | <img | 355 |
| 43 | onclick="return | 352 |
| 44 | if | 350 |
| 45 | follow:false, | 348 |
| 46 | class="hight" | 337 |
| 47 | type="hidden" | 333 |
| 48 | The | 322 |
| 49 | alt | 306 |
| 50 | class="high | 296 |

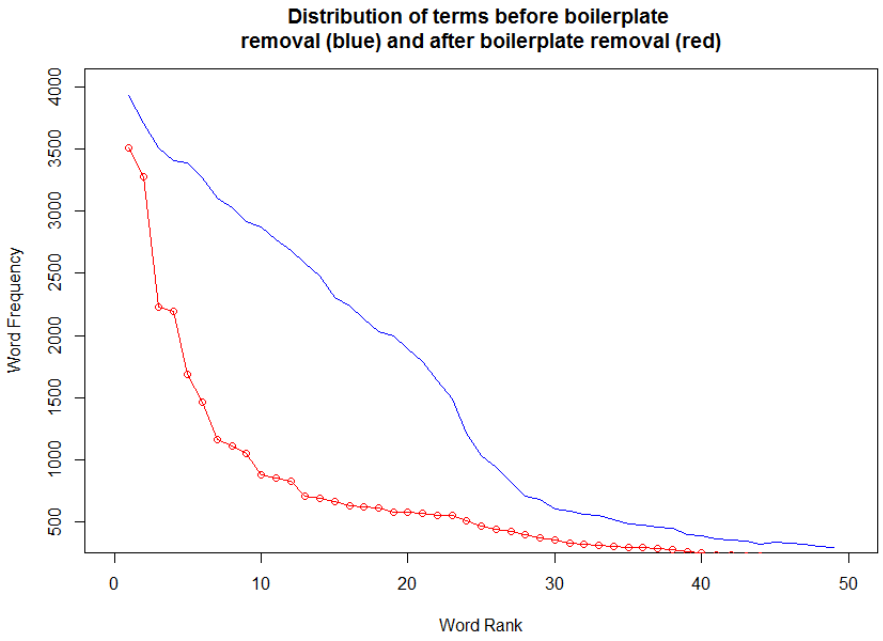


Figure 1: Graph with x-axis as word rank , and y-axis as word frequency

# Bibliography

[1] jusText. https://pypi.python.org/pypi/jusText/2.1.0.

[2] Wordcount. https://github.com/mlafeldt/google-python-class/blob/master/basic/solution/wordcount.py. Accessed: 2015-04-03.

[3] Zipf. https://plus.maths.org/content/mystery-zipf.