Words, Zipf's Law, Miller's Monkeys

Gerome Yoo

Department of Statistics, Sungkyunkwan University

2018

INDEX

Basics of English Words

Unigram Word Count

- Same as **word frequency**, when normalized
- Just simple counting of words

How do we define words?

- Words = Text Split by Space
  ex) English, Korean
- This definition is Non-Trivial for language without space
  ex) Chinese

Processing Text is called
"Tokenization" or "Text normalization"

Tokenization

Things to Consider

Throw away Junks such as Html tags!

- But sometimes they are valuable.
  ex) navigate the document structure

Word boundaries : White space and Punctuations

- What should we do with words like "Ph.D, isn't, e-mail"?
- Domain Dependent problem
- Manually created regular expression rules are typically used.

Capitalization, case-folding

- Convenient to lower case every character
  Counter example : "US" vs "us"

Tokenization

Things to Consider

Stemming(Lemmatization)

- *Stem* can be *inflected* with a morphological *suffix* to produce variation.
  ex) look $\Rightarrow$ looks,looking,looked
- Beneficial to map all inflected forms into the *stem*.
- Complex process - many exceptional cases exist.
  ex) department vs. depart

Stemming for Korean

- 동음이의어 문제 발생
  ex) 밤(밤 율) vs. 밤(밤 야),   눈(눈 목) vs. 눈(눈 설)
- Turns into disambiguation problem.

Tokenization

Things to Consider

Stopwords(불용어)

- Most frequent words often do not carry much meaning.
  ex) the, a, of, for, in $\cdots$
- Stopword list can vary from domains.
- For many NLP purposes, stopwords are nuisance.
  -Will regenerating stopwords for artificially created text seem more natural?
- Stopword removal is common preprocessing step.

TOKENIZATION

THINGS TO CONSIDER

After Tokenization

After cleaning up text, there are two concepts.

- Word Token : Occurence of a word
- Word Type : unique words
- ex) "The dog chases the cat"
  $\rightarrow$ 5 Word Tokens, 4 Word Type
  There are two tokens of word type "the".

Tokenization

Things to Consider

Vocabulary

- "Vocabulary" is list of Word Types.
- Useful to have special word type "UNK" for unknown words.

Corpus(말뭉치)

- "Corpus" is large collection of text.
  ex) several years' newspapers
- *frequency cutoff*
  Can be applied to exclude word types with small counts.
  Usually determined empirically.

Zipf's Law

Zipf's Law

The Zipf's Law is empirically known as

$$f \times r \approx constant \quad or \quad f \propto \frac{1}{r}$$

where f is word count, r is rank of the word.

- There exists a pattern,when compute $count \times rank$
  where rank is number of word types ranked by their count.
- plot $\log(r)$ on x-axis and $\log(f)$ on y-axis,
  words roughly form a line from upper-left to lower-right.
- f can be frequency(count divided by the corpus),
  the relation still hold.

ZIPF'S LAW ON "MOBYDICK"

```
>moby_stem_token <- tokenize_word_stems(mobydick)
>table2 <- table(moby_stem_token)
>table21 <- sort(table2,decreasing=T)
>table22 <- as.data.frame(table21)
>table23 <-
+cbind("rank"=as.numeric(rownames(table22)),table22)

>fr <- c()
    for(i in 1:length(table23$rank)){
      fr[i] <- (table23$rank[i])*(table23$Freq[i])
    }

>table24 <- cbind("fr"=fr,table23)
>table25 <- table24[c(3,4,2,1)]
>table25
```

Zipf's Law on "Mobydick"

```
> table25
        moby_stem_token  Freq  rank     fr
1                   the 14620     1 14620
2                    of  6736     2 13472
3                   and  6502     3 19506
4                     a  4778     4 19112
5                    to  4709     5 23545
6                    in  4231     6 25386
7                  that  3099     7 21693
8                    it  2916     8 23328
9                   his  2530     9 22770
10                    i  1989    10 19890
11                   he  1878    11 20658
12                  but  1823    12 21876
13                 with  1770    13 23010
```

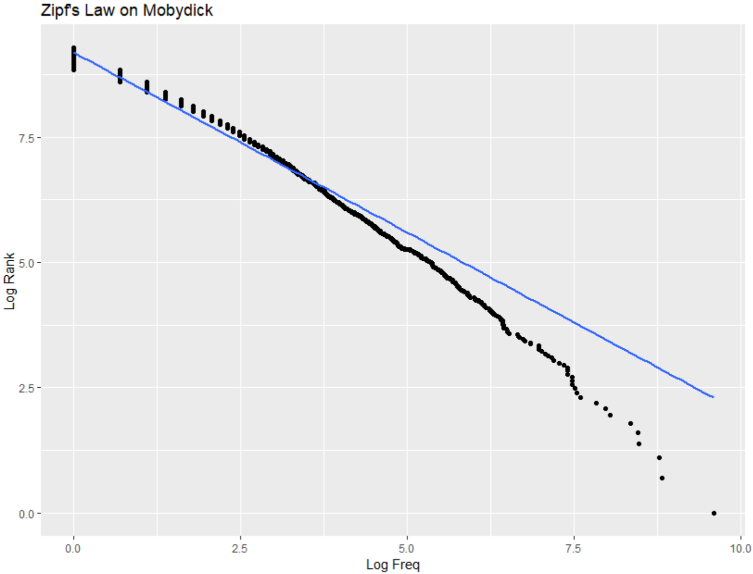
ZIPF'S LAW ON "MOBYDICK"

```
>moby_df <- table25 %>%
    mutate(log_rank = log(rank), log_f = log(Freq))

>moby_lm <- lm(log_rank~log_f, data = moby_df)
>summary(moby_lm)

>ggplot(moby_df, aes(x=`log_f`, y=log_rank)) +
  geom_point() +
  stat_smooth(method="lm", se=TRUE) +
  labs(x="Log Freq", y="Log Rank",
       title="Zipf's Law on Mobydick")
```

Zipf's Law on "Mobydick"



Zipf's Law

- See Zipf's Law R Example for more practice.

Zipf's Law

Zipf's Law generalized by Mandelbrot

The Zipf's Law generalized by Mandelbrot is

$$f = P(r + \rho)^{-B}$$

where f is word count, r is rank of the word.

- Adding more parameters to Zipf's Law.
- With more parameters, the Law become more flexible.
- See Miller's Monkey for more detail.

Miller's Monkeys

Imagine,

Promise a Monkey some stock options and ask it to type tirelessly on a computer keyboard.

What do we get?

What frequency and rank relation do monkey word possess?

Assumption for Simplication

- Keyboard has 27 keys : a to z, and white space.
- Monkey hit each key with equal probability.
- Let a sequence of letters separated by white space "Word".

Miller's Monkeys

Probability of monkey word with length i

$$P(i) = (1/27)^i(1/27) = (1/27)^{i+1}$$

- Longer the word, lower its probability and expected count
- Rank all monkey words by its probability, then

The rank r_i of a word with length i satisfies

$$\sum_{j=1}^{i-1} 26^j < r_i \leq \sum_{j=1}^{i} 26^j$$

Miller's Monkeys

Deriving 'Fractional length' i'

$$i' = \frac{log(\frac{25}{26}r + 1)}{log26}$$

proof)
Let us consider the word with rank

$$r = \sum_{j=1}^{i} 26^j = \frac{26}{25}(26^i - 1)$$

$$\frac{25}{26}r = 26^i - 1 \quad \Rightarrow \quad 26^i = \frac{25}{26}r + 1 \quad \Rightarrow$$
$$i \times log26 = log\frac{25}{26}r + 1 \quad \Rightarrow \quad i = \frac{log(\frac{25}{26}r + 1)}{log26}$$

MILLER'S MONKEYS

Word frequency with 'Fractional length' i'

$$\begin{aligned}
p(i') &= (1/27)^{i'+1} \\
&= (1/27)^{\frac{log(\frac{25}{26}r+1)}{log26}+1} \\
&= (1/27)(1/27)^{\frac{log(\frac{25}{26}r+1)}{log26}} \\
&= (1/27)(\frac{25}{26}r+1)^{\frac{log(\frac{1}{27})}{log26}} \quad \text{using the fact} \quad a^{logb} = b^{loga} \\
&= (1/27)(\frac{25}{26}r+1)^{-\frac{log27}{log26}} \\
&\approx 0.04(r+1.04)^{-1.01}
\end{aligned}$$

Miller's Monkeys

Word frequency with 'Fractional length' i'

$$\therefore \quad p(i') \quad \approx \quad 0.04(r + 1.04)^{-1.01}$$

Recall Zipf's Law generalized by Mandelbrot

$$f = P(r + \rho)^{-B}$$

$P(i')$ fits Mandelbrot's Law, and is fairly close to Zipf's Law.

Zipf's law may not reflect some deep knowledge of Language.
It still points to an important observation that

"Almost all words are rare!"

Reference

Just reorganization of

- CS838-1 Advanced NLP class material by Xiaojin Zhu
- Introduction to the tokenizers Package
- Opensource Shakespeare
- NLTK 자연어 처리 패키지