Comp 540: Information Retrieval **Fall 2013**

Assignment 1 — November 1, 2013

Osman Başkaya Lecturer: Emine Yilmaz

Answers

Answer 1-a Total number of tokens: 26693, unique number of words: 2632

It shows the Zipf's Law.

Words, probabilities, ranks and other relevant information can be found in *alice-25-most-freq-starts-with-f.txt* and *alice-25-most-freq.txt*. Code that related to this answer can be found in *Makefile*. Please note that I used basic UNIX commands such as **grep, wc, head, tail** etc. for this question.

Answer 1-b According to Zipf's law related to proportion, we will omit $k/2 + k/3 + k/4 + k/5$, in total, (1 - k*77/60)% words are omitted. k is approximately 0.1 for English. If we plug k as 0.1 we get 0.128% tokens are omitted. Additionally, actual omitted proportion will be $(26692 - 23437)/26692$, results in 0.1219% Note that this is the number of tokens and it's strongly correlated with the theory.

Answer 2 For question 2, I found m, b equal to -1.28796563313, 7.22527323685, respectively. The script named *find-zipf-line.py* finds the parameter of the line and draws the line and the data.

If we find the k from here we will find $e^b = exp\{7.225\}$.

Answer 3 According to Heaps' Law, we have this equation:

$$\frac{V_r(n_1)}{Kn_1^\beta} = \frac{V_r(n)}{Kn^\beta} \tag{1.1}$$

90% of vocabulary is read so we can replace $V_r(n_1)$ with 9 and $V_r(n)$ with 10. We also know that n equals 26693. If we run the script named *heaps-law.py* for the first part (a), we will get the 0.81%.

```
./heaps-law.py a alice.parsed.txt 0.9 0.5
0.81 21621.33
```

Heaps' Law says that if we want to come across 90% of vocabulary, we need approximately 21621 tokens in our case which means the proportion is 0.81%.

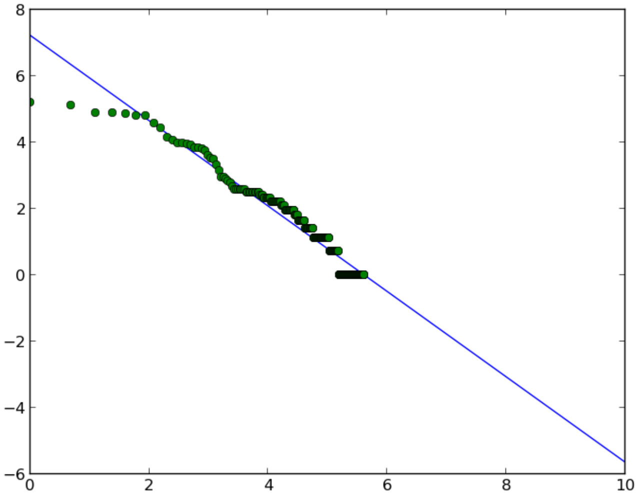

Figure 1.1. Log-log plot of the data. Blue line demonstrates the best line and green dots from the data

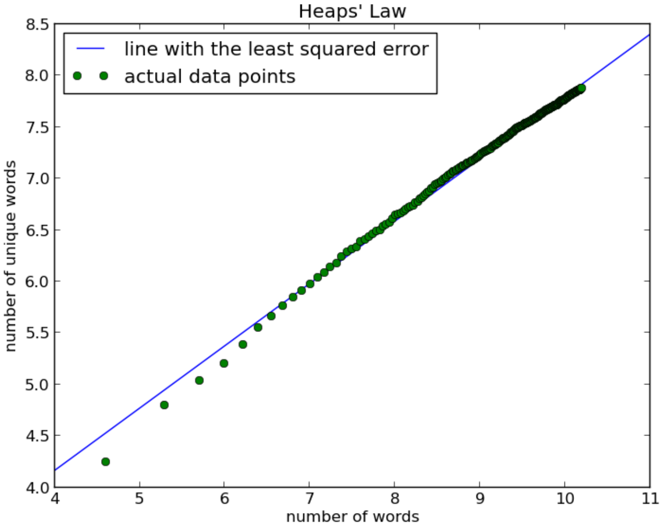

Figure 1.2. Log-log plot of the data. Blue line demonstrates the best line and green dots from the data

For the second part I used the formula given for question 2. First I posed the problem as line fitting. If you run the script named *heaps-law.py*, it calculates β and k, as well as it draws a figure that contains best fitting line and the unique words-words pairs.

```
./heaps-law.py b alice.parsed.txt
beta: 0.613285692043, k: 5.31979957386
```

End of File