AmirMohammad KahooriZade
Information Retrieval
610397178

Heaps and zipfs law

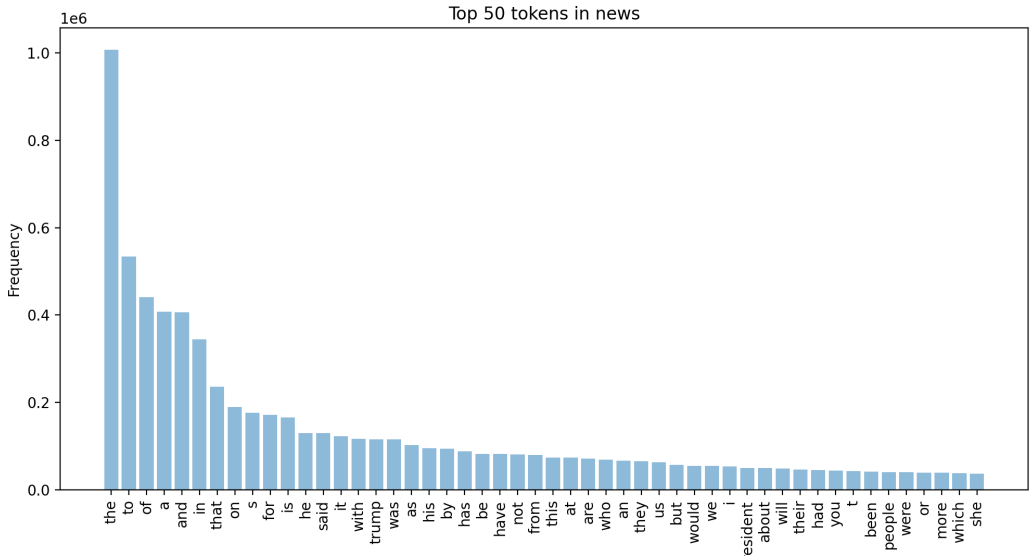
1. Datasets

    1. English news(https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset)

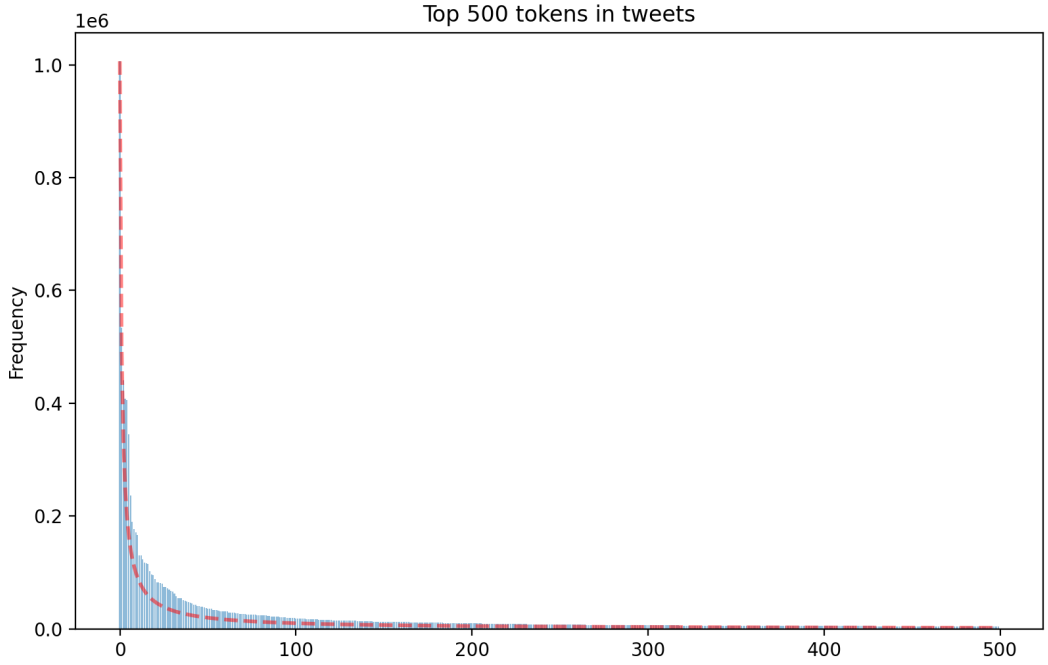
    2. Persian news(https://bigdata-ir.com/wp-content/uploads/2019/07/news-dataset.zip)

2. General

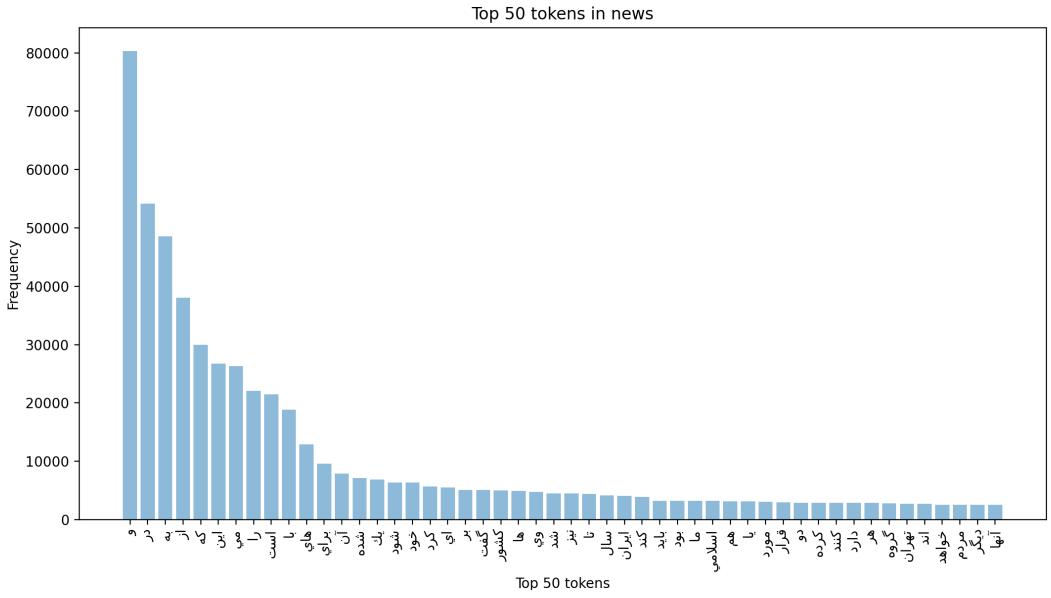
    • English dataset has around 18000000 words, the chart below shows 50 most frequent words(as you can see trump is the 16 frequent word and thats funny:))



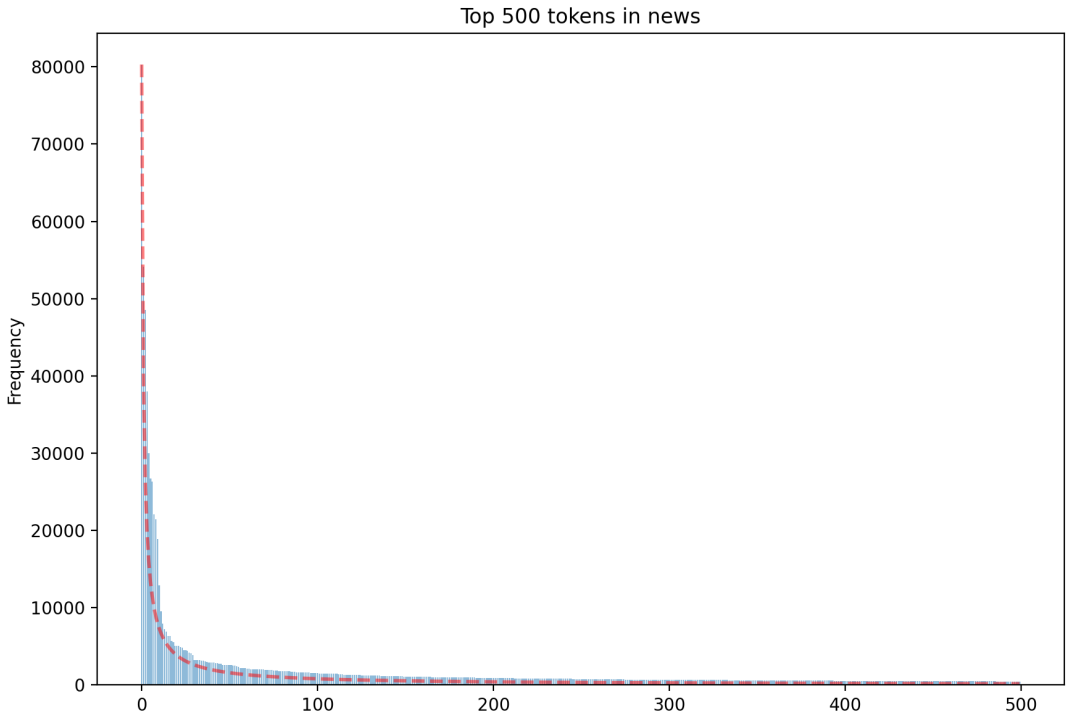
And if you wonder how does a Barchart looks like for a dataset this size here is a taste of that, I selected top 500 words, as you can see if I had selected all the words you couldn't see any thing!



• Now some general stats for Persian dataset, it has around 1600000 words and you can see top 50 chart below

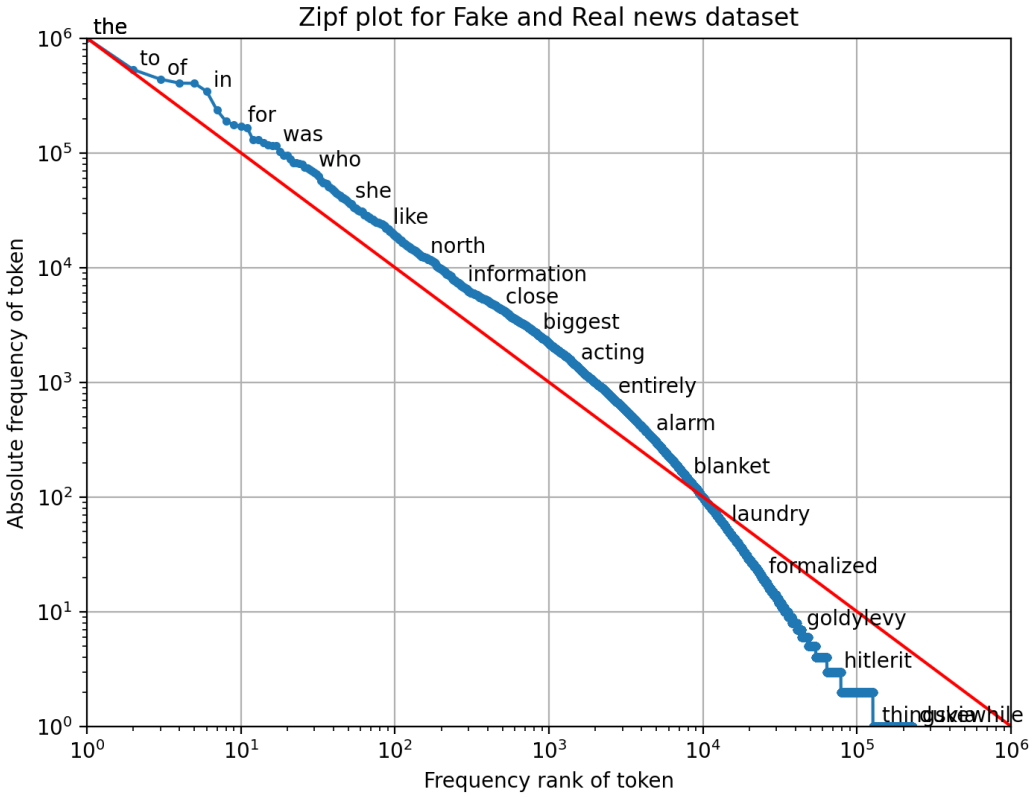


And the frequency Barchart for top 500 words, as you can see, looks pretty much like the English chart,

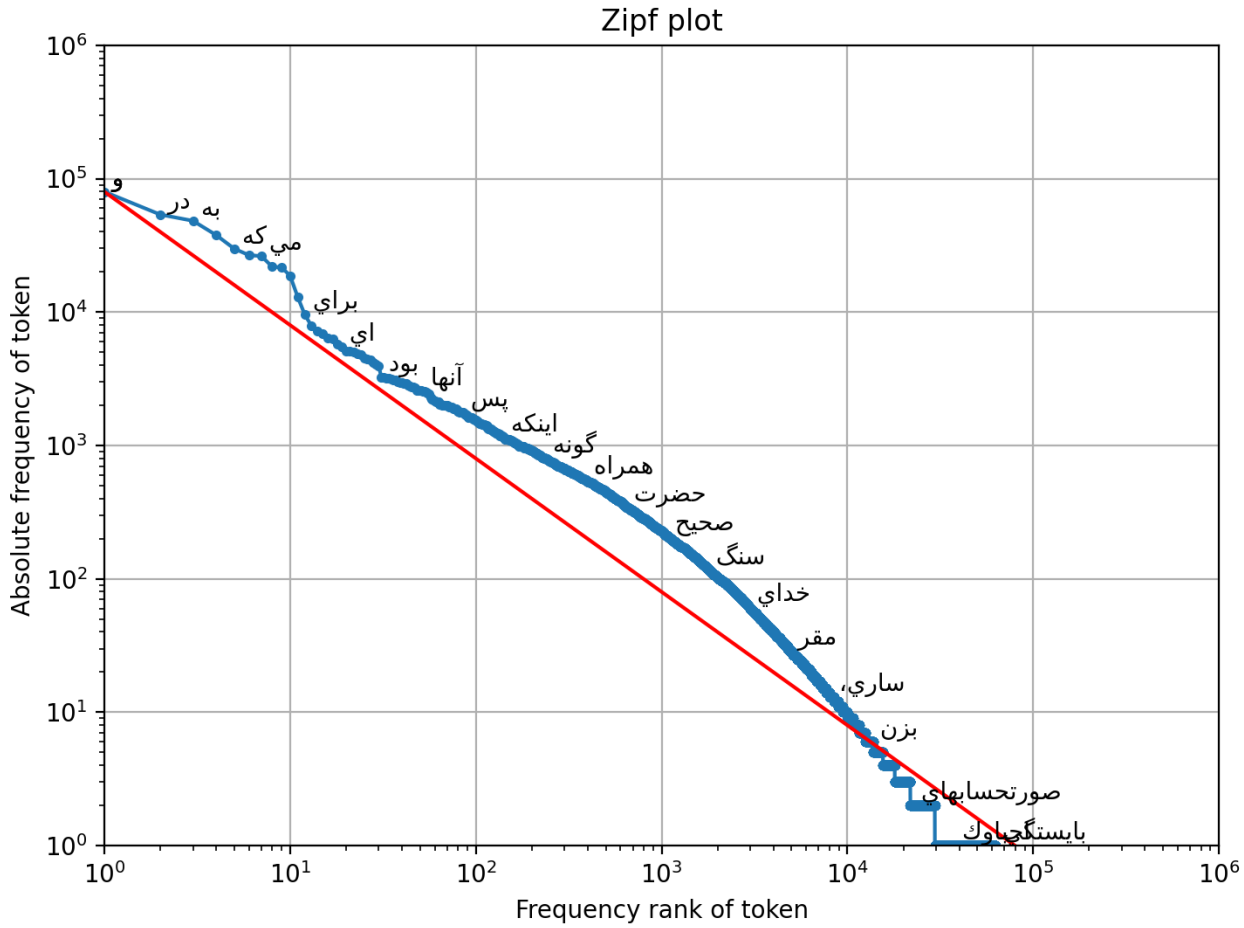


3. Zipf's law

Zipf's law was pretty simple, the prediction line is obvious and I've ploted the observed line with little dots like this (rank, frequency) you can see the final chart for English dataset below



Now the Persian dataset:



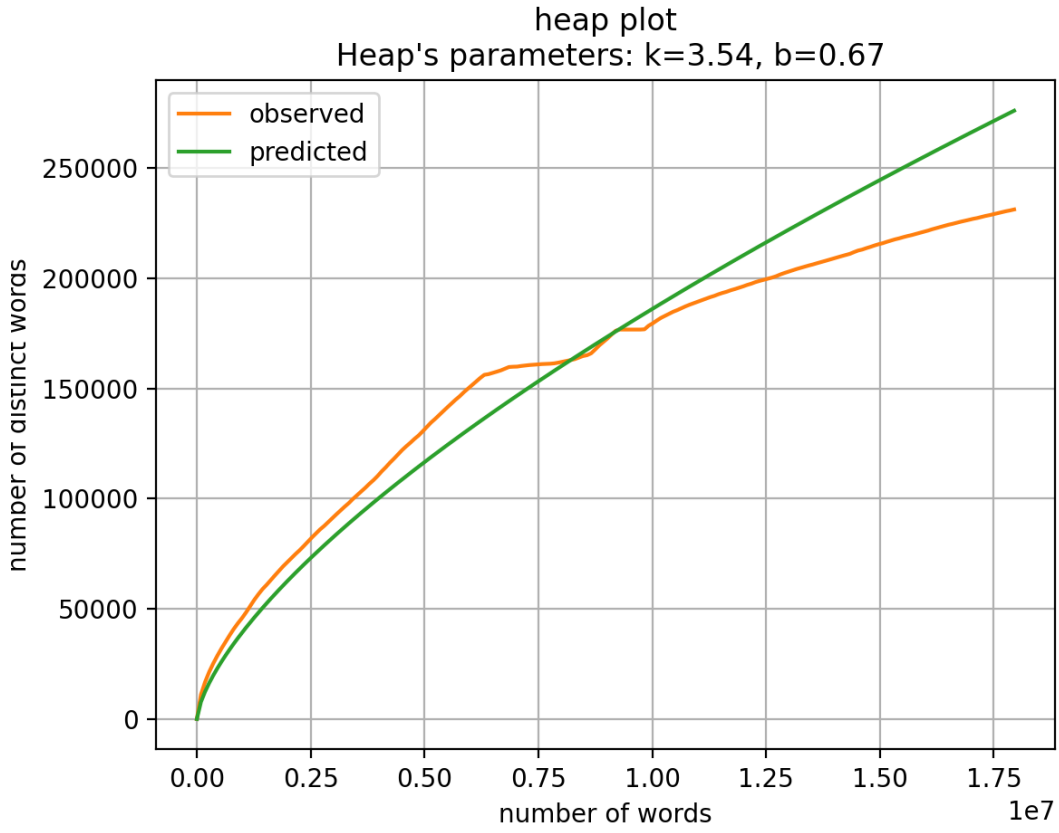
4. Heap's law

The heap's law was a little bit challenging for me, at first I thought it's as easy as it gets but then after some issues I realized I was to far gone! So I started a new approach, for every 200 words the distinct number of words would be calculated and we use that to draw the chart and find parameters k, b.

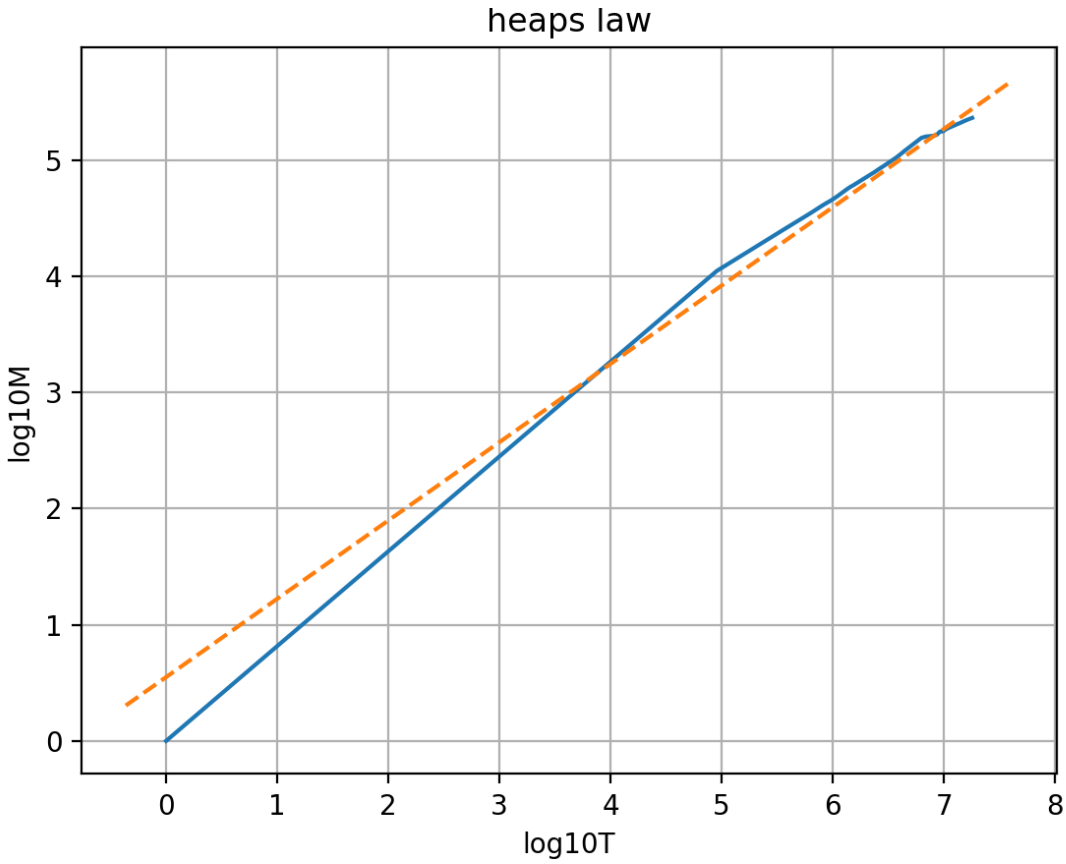
I've used linear regression algorithm to line up the chart perfectly.

For heap's law I've generated two plots, one is just the number of words vs number of distinct words, no logs on any of them, and the other one is a log log plot that you can see the linearity concept of heap's law, but the first plot puts thing in the perspective in my opinion.

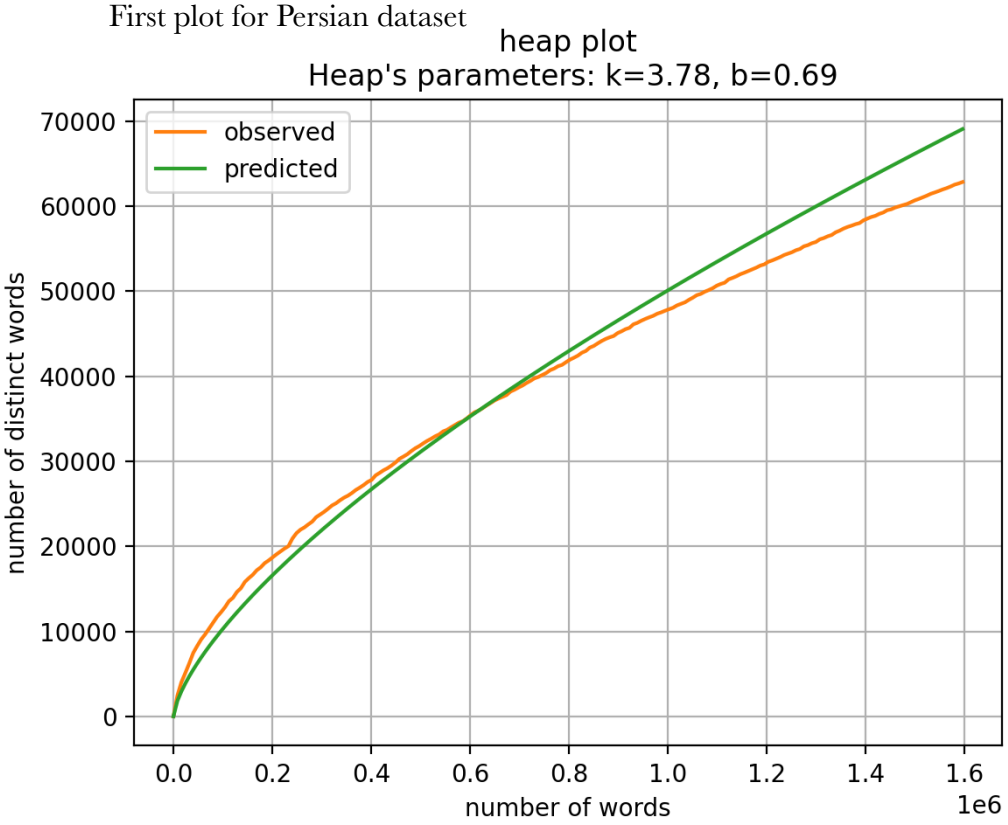
First plot for the English dataset:



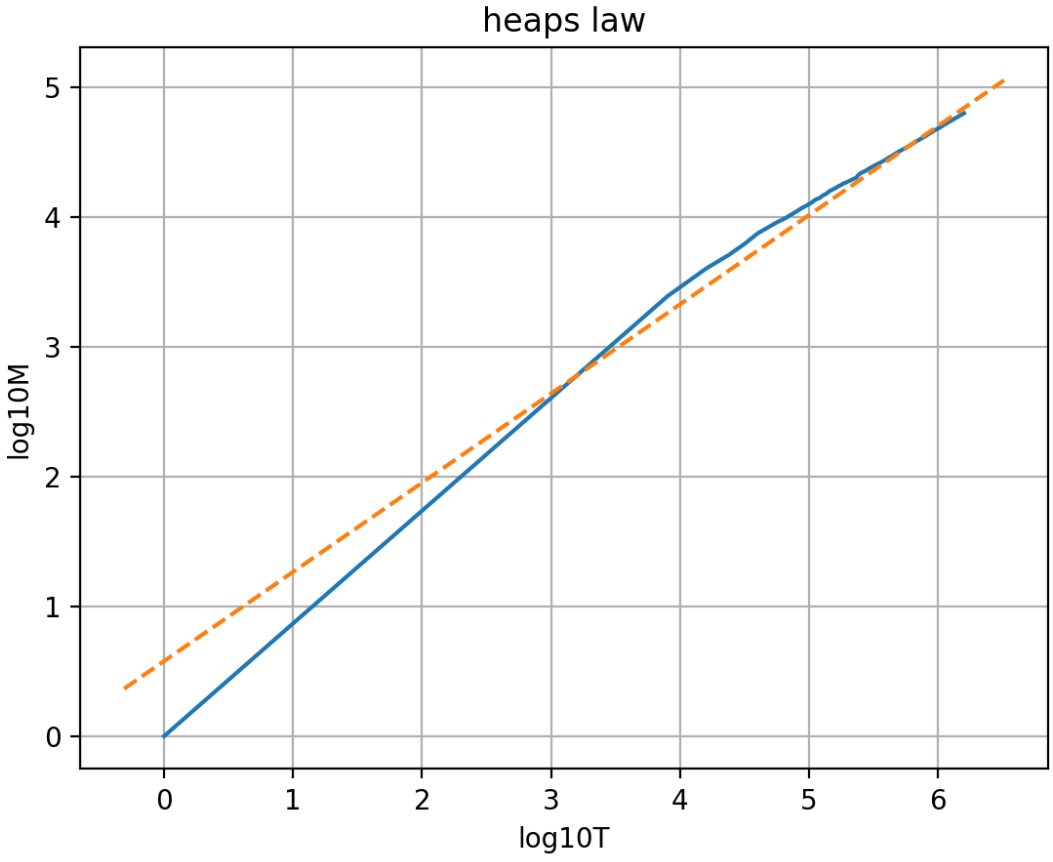
Loglog plot for English dataset, k=3.54, b=0.67



First plot for Persian dataset

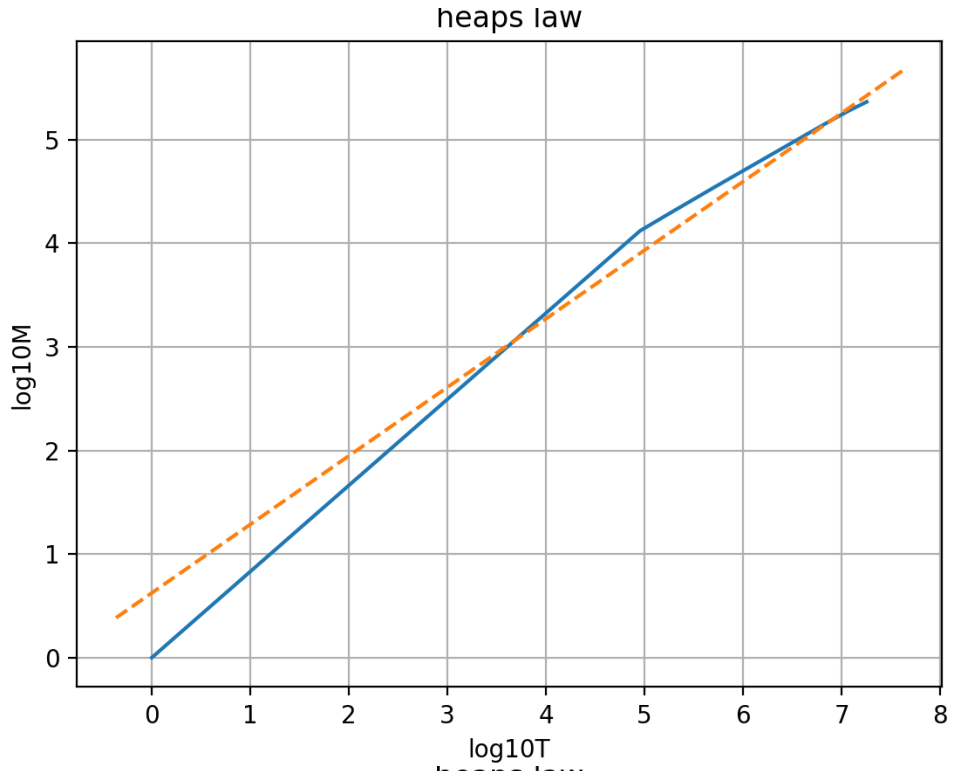


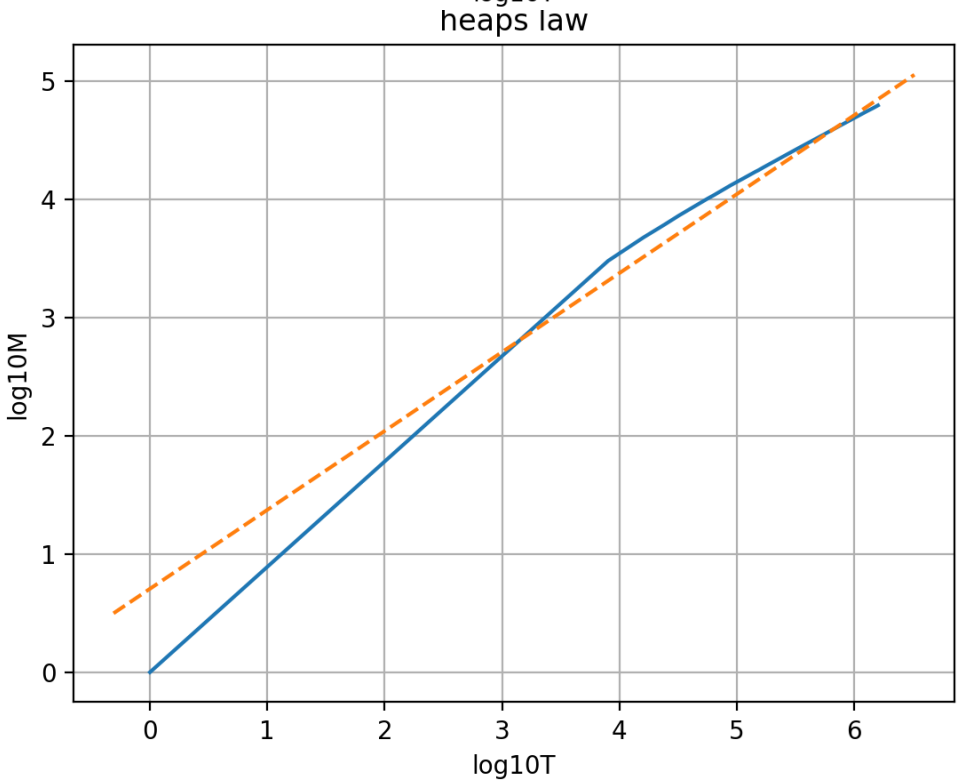
Loglog Plot for Persian dataset, k=3.78, b=0.69

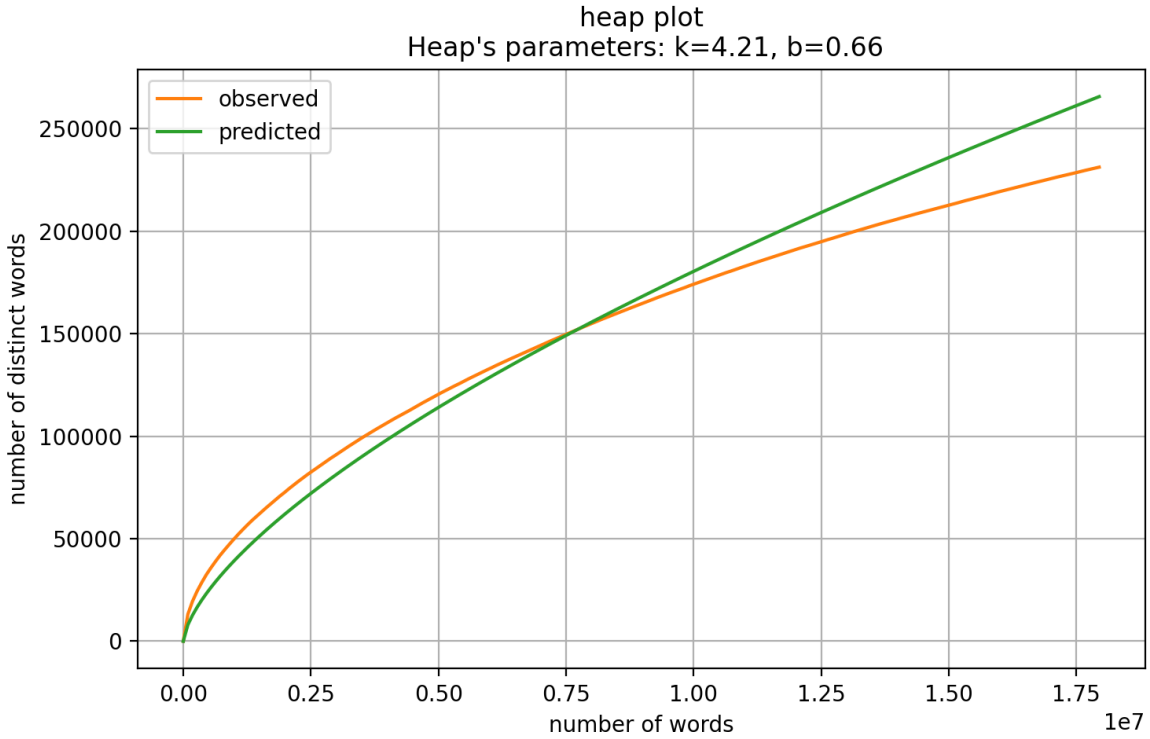


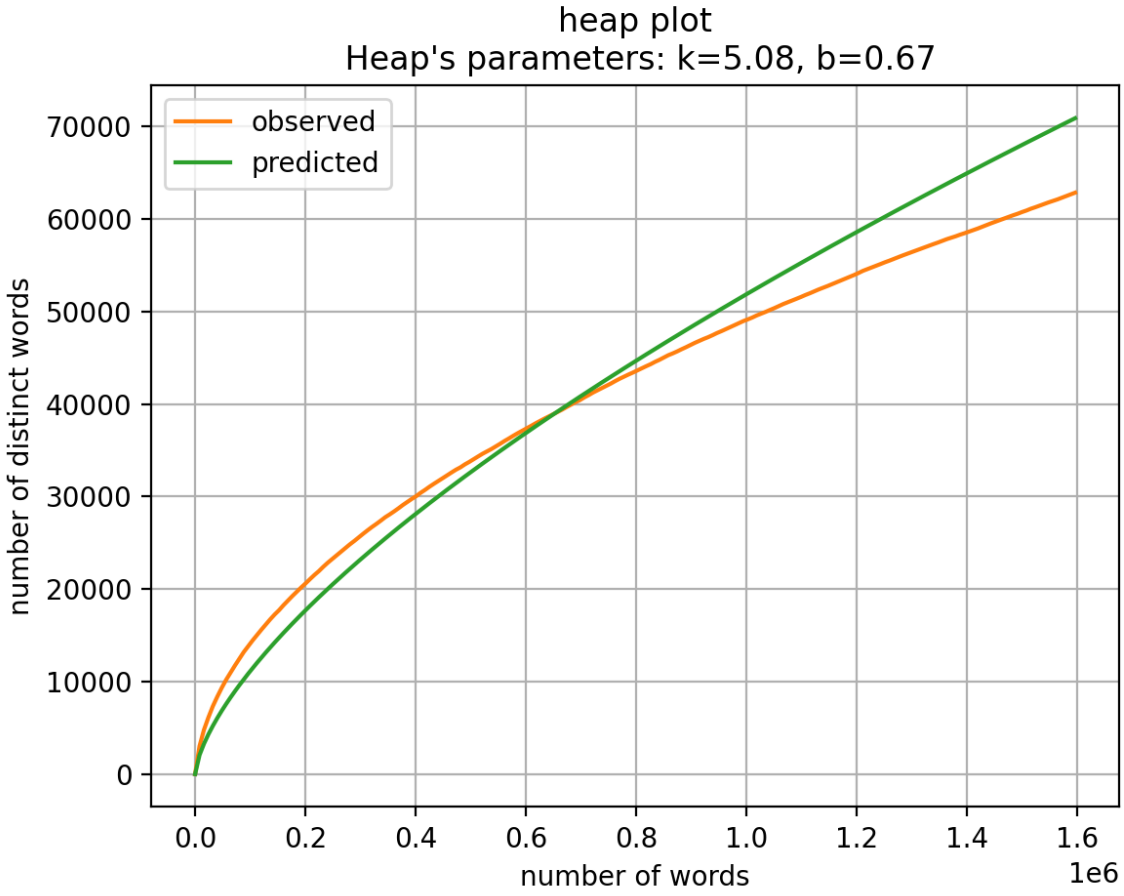
I had a fun experiment, after collecting all the words I said lets shuffle them all and then test the heap's law, what I found was amusing to me, both blots for both datasets were almost identical!(with different numbers of course!)

Here are the charts, and to make it fun I will not name the charts so you can guess them.






heap plot
Heap's parameters: k=4.21, b=0.66
observed
predicted
number of distinct words
250000
200000
150000
100000
50000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
number of words
1e7


heap plot
Heap's parameters: k=5.08, b=0.67
observed
predicted
number of distinct words
70000
60000
50000
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
number of words
1e6

5. What is the effect of including spelling errors VS, automatically correcting spelling errors on heaps' law?

Well I haven't actually tested it but I think there is two type of correction here, there are some unintentional spelling errors, if we don't correct them there will be an increase in number of distinct words but its not that big of a deal and nothing will happen in heaps law, but there are some intentional miss spells that if we correct them I do think we will have a problem, I mean if we start analyzing every single article that exists at some point we would be out of words that the dictionary says are legitimate, but its the creativity of us humans that will generate new words and thats how the heaps law keeps up after seeing 30 trillion words or more.

6. Compute the vocabulary of size M for this scenario:

• Looking at a collection of web pages, you find that there are 3000 different terms in the first 10000 tokens, and 30000 terms in the first 1000000 tokens.

• Assume a search engine indexes a total of 20000000000 (2×10^{10}) web pages containing 200 tokens on average.

• What is the size of the vocabulary of the indexed collection as predicted by heaps law?

As we know in heaps law b is roughly around 0.5, with that assumption we can say that k =30, so with a total word size of 44×10^{12} we would have around 6×10^{7} terms.