Lab 1 of
Information Retrieval (IR)

FangYuan Zhao
Qiaorui Xiang

1 Zipf's Law

The purpose of this section is to look at the word distribution in *novels* data, and see if Zipf's law holds, which is to check if the rank-frequency seems to follow a power law distribution.

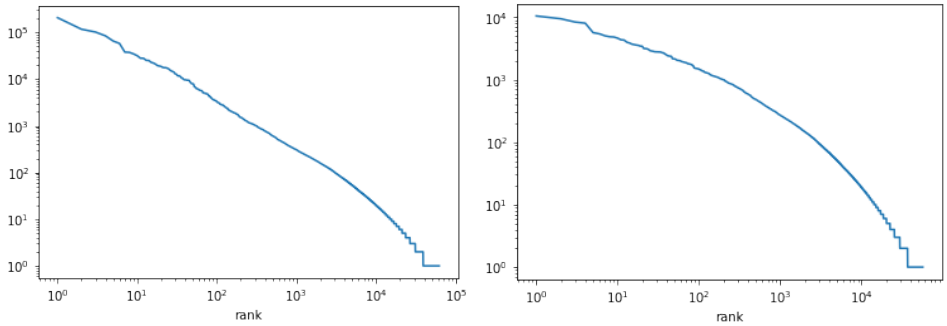

Figure 1: Word frequency before (right) and after (left) removing noises

We used modified version of **IndexFiles.py** in order to accept multiple file input. Then we compute the word frequency by a piece of code that we take from **CountWords.py**. Some noises have been observed, so we take out words that are either just numbers, url's, binary or unreadables, dates, and sorted the list again. The result is shown in Figure 1, both plots seem to follow a power law distribution.

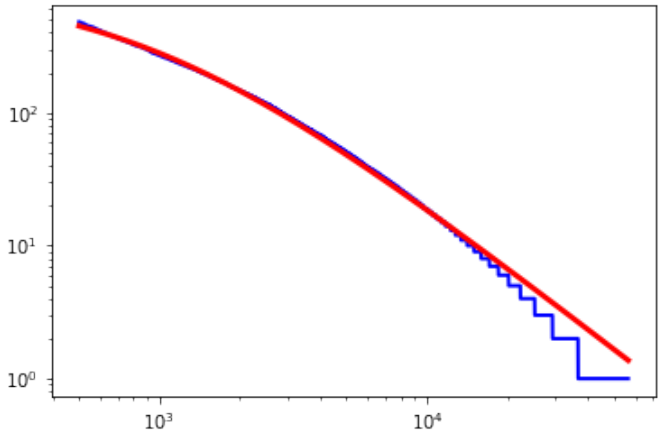

Figure 2: Zipf's law with filtering out highest 500 ranks

However, we observe a curve after removing noises due to high influences from high ranks. So when we perform **curve fitting** to get the parameters, we filter out 500 highest ranks. The result of curve fitting (using **curve_fit** function) give us best value for our parameters:

$$\alpha = 1.5678423947288782$$

$$b = 927.5702358034353$$

$$c = 39723061.62341447$$

We plot the function with those parameters and compare with data as shown in Figure 2. We can see that Zipf's law approximation which is the red line, fits well with our data which is the blue line.

2 Heap's Law

This exercise is to check if Heaps' law can represent the number of distinct terms in a piece of text with some k and β.

In order to generate sample data, we create indices containing different number of novels. We have in total 33 documents, so we perform 33 iterations. For each iteration i we generate a index with i documents. Then we store the total number of words in each index, and the number of different words in each.

Finally we did same as previous section. We use **curve_fit** function to perform curve fitting got best value for our parameters:

$$k = 9.230328206591347$$

$$\beta = 0.6141579887156269$$

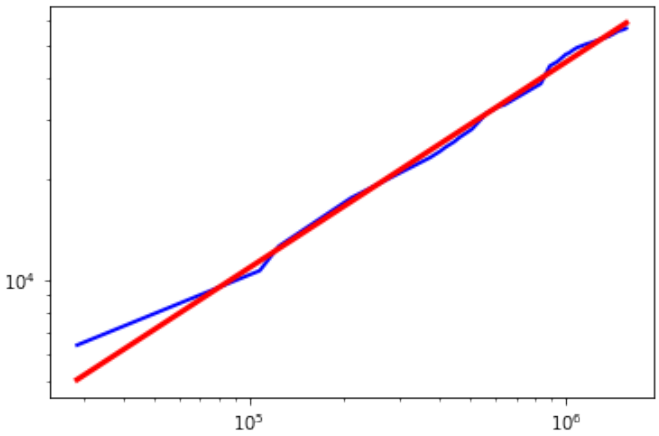

Figure 3: Heaps' law

We can see from Figure 3 above, the Heaps's law approximation (red line) fit pretty well with the sample (blue line).

3 Final Thoughts

We can conclude that both Zipf's law and Heaps' law did hold on the *novels* dataset since we were able to find parameters that fit to our data.