Experiment 1
Word Cloud, Zipf's law, Map-Reduce

I. Experimentation Done

1) Verification of Zipf's law from a classic from Gutenberg corpus.
2) Performed stemming stop word removal and visualized the difference through Word Cloud.
3) Configure Hadoop and map-reduce.

II. Approach

A. Environment Setup

1) We downloaded a classic from Gutenberg Corpus. Imported various python libraries such as numpy, pandas, nltk(natural language toolkit) and re(for regular expression). Performed tokenization and word count on the classic.
2) Map Reducer on Hadoop: We installed Hadoop and completed setting up a config file. Here, one machine in the cluster is designated as the Name Node and another machine as the Resource cluster.
3) It involves unpacking the software on all the machines in the cluster or installing it through a packaging system whatever is suitable for your operating system and Manager.These are the masters.
4) Other services are usually run either on dedicated hardware or on shared infrastructure, depending upon the load. The rest of the machines in the cluster act as both Data Node and Node Manager. These are the workers.

B. Data pre-processing

1) Tokeinzation :- We break our text downloaded from the Guttenberg corpus into individual tokens or words so that we can create theri frequency and ranks for verification of Zipf's law.
2) Ranking Tokens:- Then we ranked the words in two different ways, first through "rankdata" a function of scipy library and in the second approach, we ranked the data through python enumeration.

C. Verification of Zipf's law

Then we plotted the frequency v/s rank taking the log of both and analyzed the plots generated for both the ranking techniques as we described above.

D. Word Cloud generation

Next, we analyzed the occurrence of words in our text using word cloud and at this stage, we have not performed any stopwords removal or stemming. Then we again visualized our text through word cloud but this time we performed stopword removal and in the next step we also performed stemming along with stopword removal to analyze the effect of stemming.

Finally, we configured Hadoop on our local machine and with the help of map-reduce performed word count and verified Zipf's law once again.

III. Discussion on the Result

We were able to verify Zipf's law from both approaches of ranking. In the verification of Zipf's law when we plotted the log of rank against the log of frequency of words we found that the plots obtained from the two different approaches of the ranking were different for the same corpus.

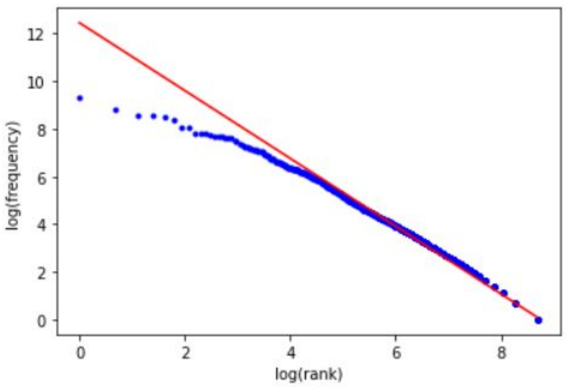


fig 1.1

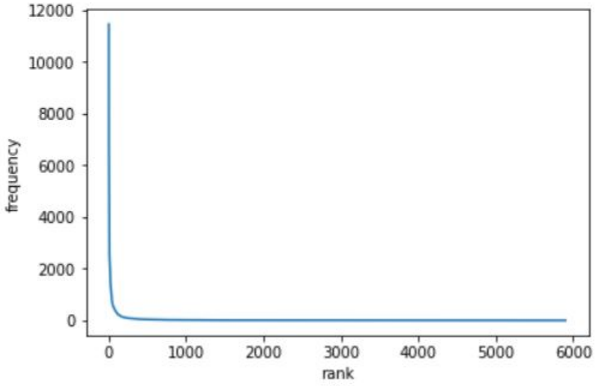


fig 1.2

log(rank) v/s log(freq) plot(fig 1.1) and rank v/s frequency plot(fig 1.2) for ranking created using RankData function

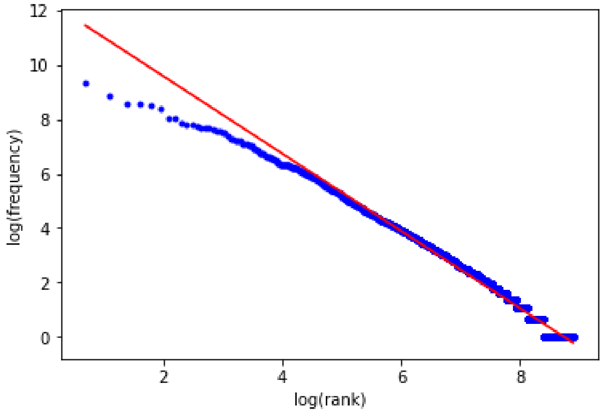


fig 1.3

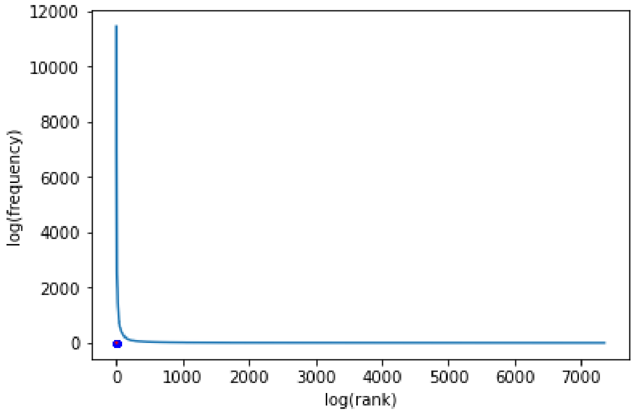
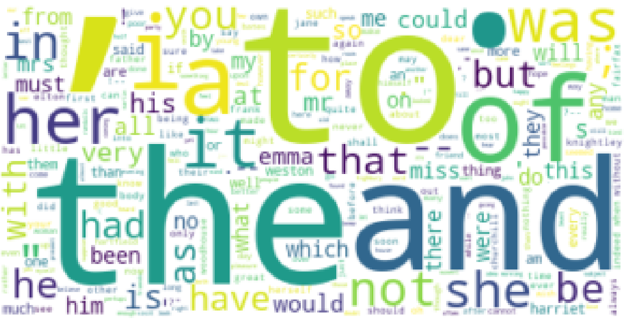


fig 1.4

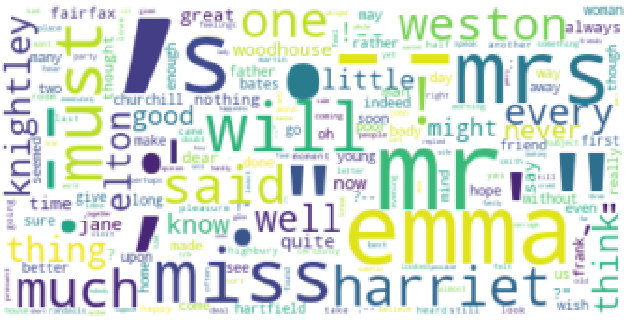
log(rank) v/s log(freq) plot(fig 1.1) and rank v/s frequency plot(fig 1.2) for ranking created using python enumerations

Visualization of our text through word cloud revealed that the common word of English dictionary such as a, and, she which are known as stop words were more frequent. The same was seen when we ranked our words and these words got higher ranks.

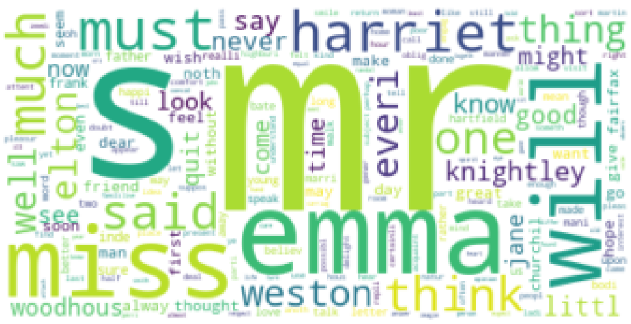
After the removal of stopwords, we were able to get a better visualization of our text and stemming improved our representation of the text as word cloud further.



Word Cloud Without removal of Stop Words



Word Cloud after removal of Stop Words



Word Cloud after removal of Stop Words and stemming

IV. Interpretation

During verification of Zipf's law, we observed that the plots for log of rank vs log of frequency were different for the two approaches of ranking. This was observed as the enumeration strategy to rank the words gives different ranks to words having the same frequency but the "rankdata" function of scipy took care of this scenario and assigned the words having the same frequency the same rank.

During the word cloud representation of the text, we found the stopwords to be dominant and this was also observed in the ranking as stop words had lower numerical ranks. On removing stopwords from our text and removing unwanted symbols we had a better picture of our text in terms of most frequent words. The stopwords are mainly for grammatical purposes and don't carry any semantic meaning for us to interpret, so for a better understanding of the text, they can be removed. Then we performed stemming on our tokenized text which enhanced the visualization in terms of the semantics of the text. Stemming enhances our visualization and understanding of the text as it tries to take the words to its root word removing grammatical structure so that we have a better picture in terms of the meaning that they carry.

V. Conclusion

Stopwords dominate our texts in terms of frequency and in accordance with Zipf's law they have lower numerical rank. They don't carry any semantic importance to our text and can be removed before further analysis of the texts. To further enhance the understanding of the text, stemming is a good approach as it tries to reduce words to the same root but it is a heuristic approach as it just chops off words without any morphological analysis.