UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

INFORMATION RETRIEVAL AND RECOMMENDER SYSTEMS

ElasticSearch and Zipf 's and Heaps' laws

Carol Jhasmin Azparrent Estela
Mohana Fathollahi

September 27, 2022

1 Preprocessing

To do preprocessing step in corpus we selected only those that are words with the following expression:

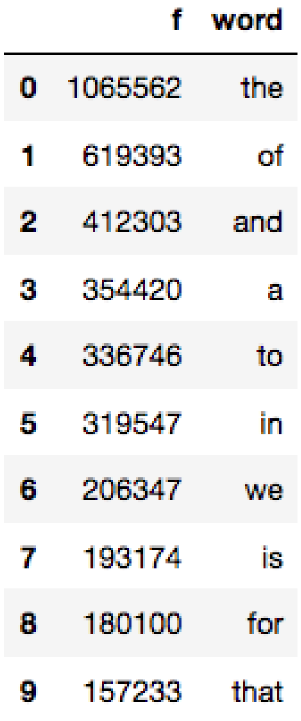
```
'(?<!\S)[A–Za–z]+(?!\S)|(?<!\S)[A–Za–z]+(?=:(?!\S))'
```

Figure 1: Pattern for preproccesing

For this section, we have run the following lines in the terminal in order to get the frequency and the rank of each word.

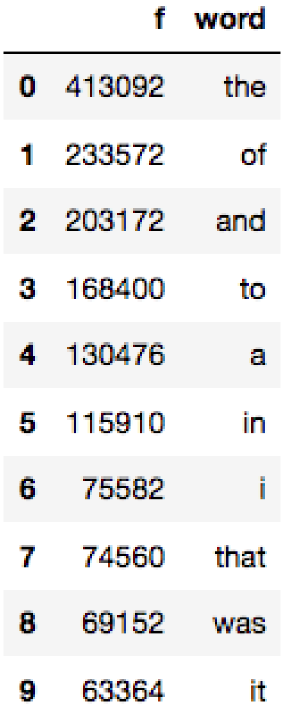
```
$ python IndexFiles.py —index news —path /carol/Documents/db/20_newsgroup
$ python CountWords.py —index news
```

This was for `20_newsgroup` but we did the same for `novels` and `arxiv_abs`. As a consequence, we got the next figures for each files. They show the rank, frequency and word in descending order based on frequency for novel, news and the arxiv articles. These tables display 10 first rows of each file. As we can see, frequency of "the" in three scripts is higher than other words but for other words we do not have same pattern. Another thing that is common in these three scripts is that 10 most frequent words will not give any relevant information due to the fact that these are articles without any relevant meaning.



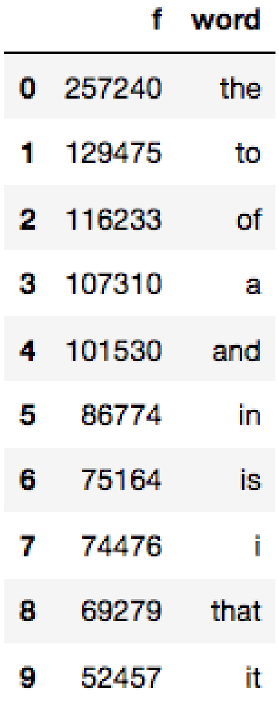
	f	word
0	1065562	the
1	619393	of
2	412303	and
3	354420	a
4	336746	to
5	319547	in
6	206347	we
7	193174	is
8	180100	for
9	157233	that

Figure 2: Arxiv



	f	word
0	413092	the
1	233572	of
2	203172	and
3	168400	to
4	130476	a
5	115910	in
6	75582	i
7	74560	that
8	69152	was
9	63364	it

Figure 3: Novel



	f	word
0	257240	the
1	129475	to
2	116233	of
3	107310	a
4	101530	and
5	86774	in
6	75164	is
7	74476	i
8	69279	that
9	52457	it

Figure 4: News

2 Zipf's Law

In this part based on information we get from frequency and rank of word we can check if our data follow Zipf's law or not.[1]

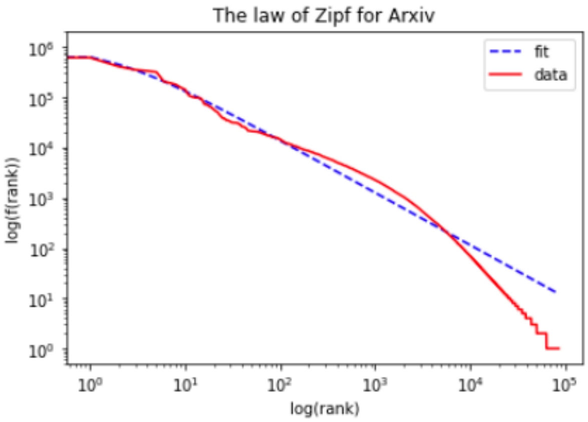


Figure 5: Zipf's law for Arxiv

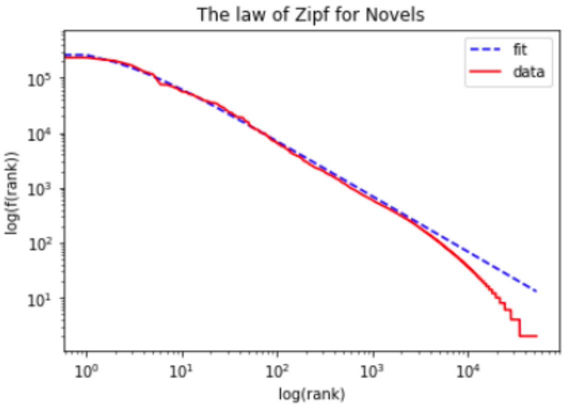


Figure 6: Zipf's law for Novels

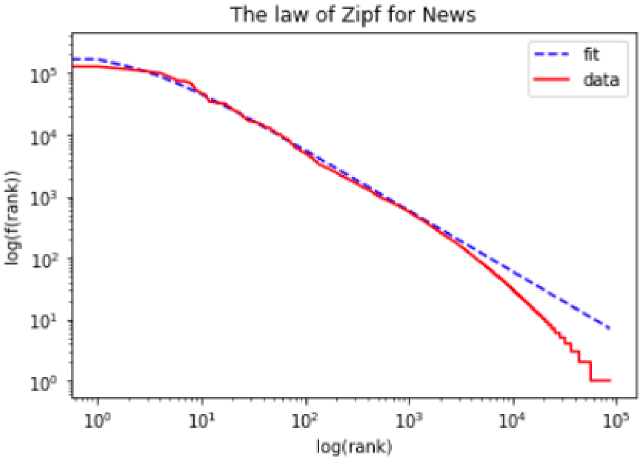


Figure 7: Zipf's law for News

As we discussed in first part of report, we have most frequent words that are kind of noisy and for that we removed first 10 words that have higher frequency, consider it as first approach. Another problem that we have in all three figures above is that we could not predict words with lowest frequency, for this problem we divided words in 2 parts, first half has half of words that has highest frequency(second approach) and second half is words with lowest frequency (third approach). In figure below behaviour of these methods for novels corpus have been shown.

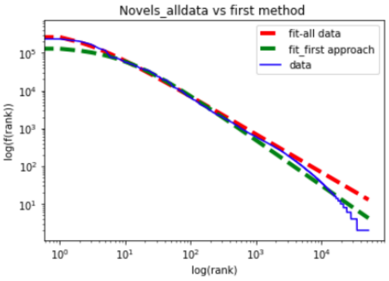


Figure 8: First comparison

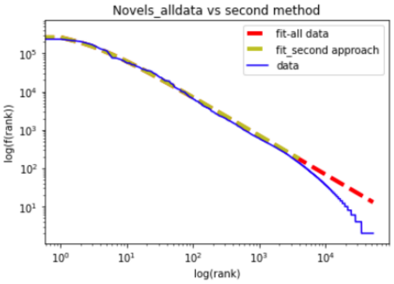


Figure 9: Second Comparison

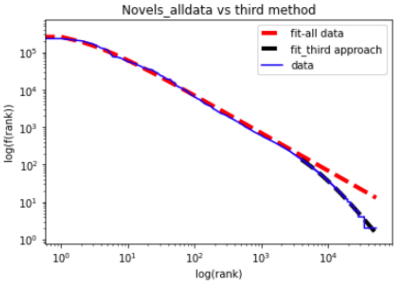


Figure 10: Third Comparison

We got approximately same behaviour for news too but for Arxiv corpus we got different result that is shown in figure below. As we can see the second approach does not behave well on this corpus.

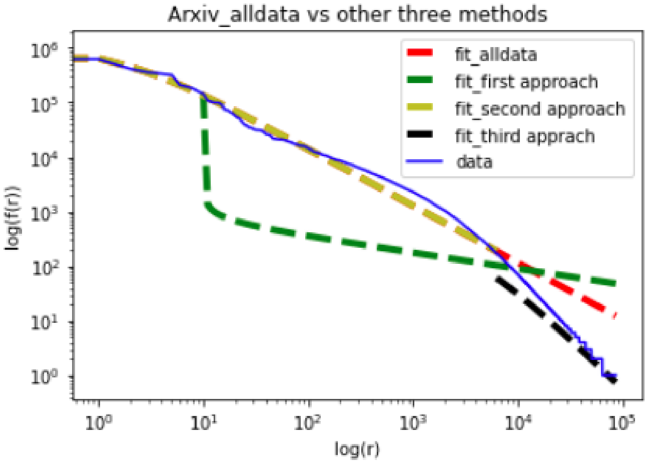


Figure 11: Comparison in Arxiv

In the table below we obtaind a, b and c for three corpus and for last two methods that have been used.

Table 1: Obtain value for a, b and c

Index	**Method**	**a**	**b**	**c**
Novel	fit_second approach	1	1.79	$7.31 * 10^5$
Novel	fit_third approach	2.29	$3.74 * 10^3$	$1.12 * 10^11$
News	fit_second approach	0.99	2.28	$5.42 * 10^5$
News	fit_third approach	1.89	$2.27 * 10^3$	$1.61 * 10^9$
Arxiv	fit_second approach	1.04	1.58	$1.71 * 10^6$
Arxiv	fit_third approach	2.15	$7.83 * 10^2$	$3.14 * 10^10$

2.1 Heaps' Law

In heap's law we have this formula: $d = k * n^{\beta}$, different corpus have been used to find the values of k and β. For this reason we used 2 approaches:

2.1.1 First approach

Randomly generated subset of novels.In this approach we applied some modification on IndexFiles.py and CountWords.py to create subset of novels. In this method we found that k = 54.24 and $\beta = 0.458$, and in the figure below we can see the behaviour of this approach.

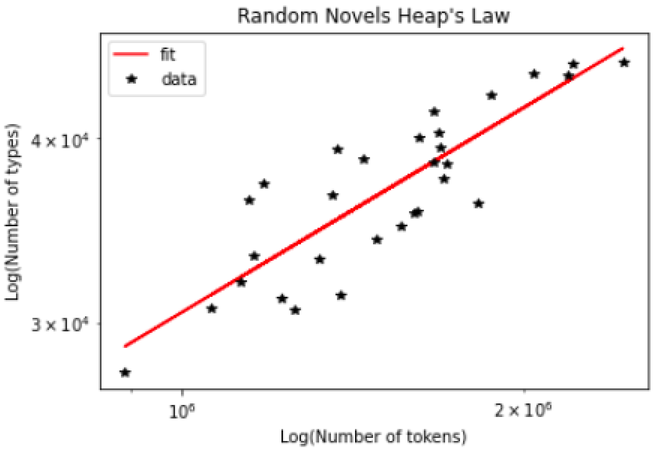


Figure 12: Novels Heaps' Law by randomness

2.1.2 Second approach

In this approach the classified of 33 novels by topic has been used.
We obtained: $k = 4.55$ and $\beta = 0.64$. In Figure below, we can see that there is a less variance compare to randomly generated case, additionally we have smaller range of k and β.

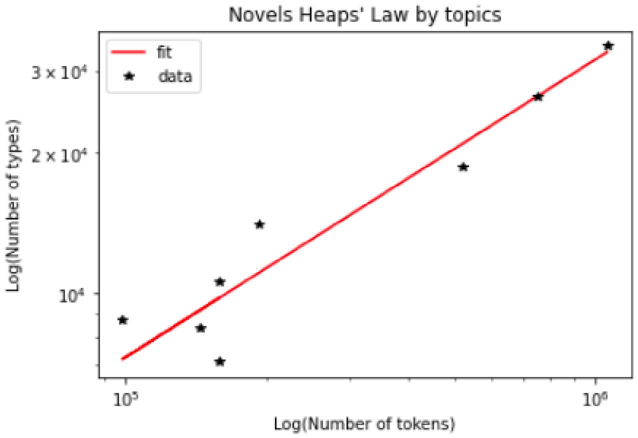


Figure 13: Novels Heaps' Law by topics

References

[1] A. Hernández-Fernández and R.F. Cancho. *Lingüıstica cuantitativa: la estadıstica de las palabras*. Grandes ideas de las matemáticas. Emse Edapp, S.L., 2019. ISBN: 9788417811884. URL: `https://books.google.es/books?id=0SdezQEACAAJ`.