Authors: Yann Lafaille & Théo Bekhedda **10/10/2021**

Information retrieval - First lab report : ElasticSearch and Zip's and Heap's laws

Zipf's law

Summary

The purpose of this first part was to find Zipf's law. We recall that the Zipf's law allows to obtain the frequency ***f*** of occurrences of a word from its ***rank*** in the text thanks to the following relationship :

$$f = \frac{c}{(rank+b)^{a}}$$

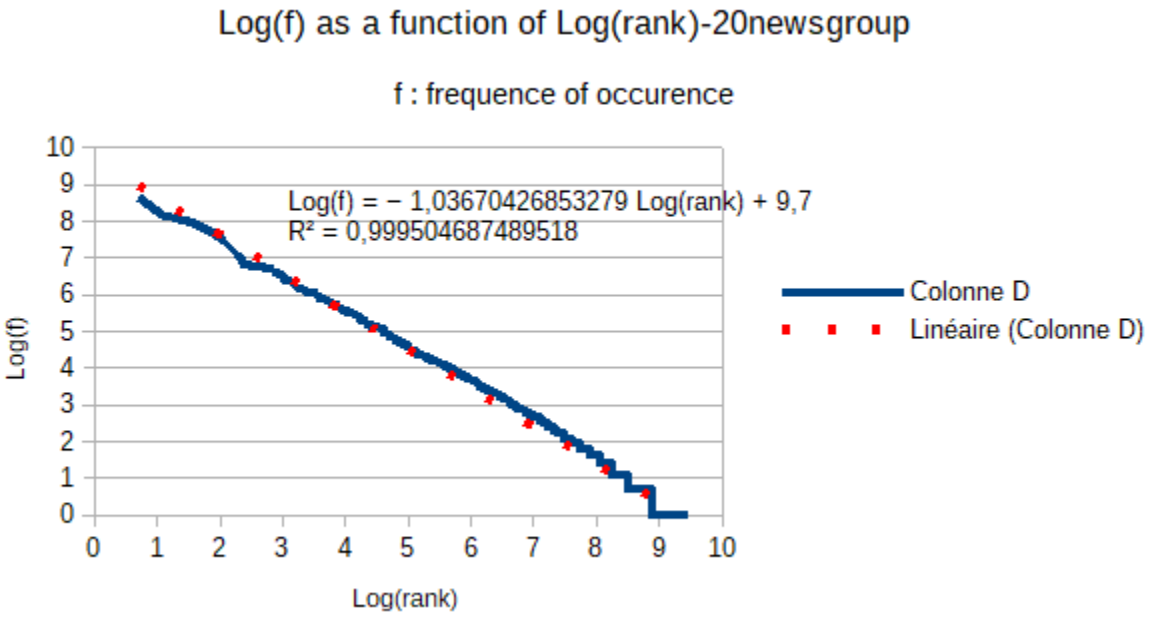
We are going to try to determine the best value for constants a,b et c.

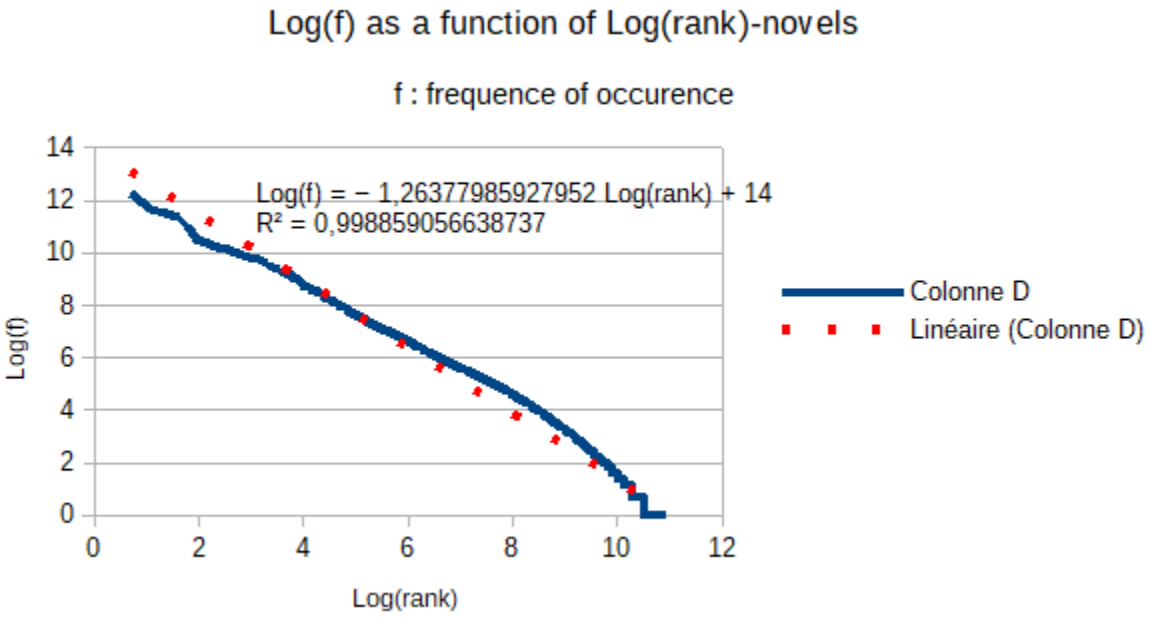
In order to do that, we created 3 indexes in ElasticSearch and for each of them, thanks to the Countwords script provided to us in python, we were able to get the list of words of each index with their respective frequency of occurrences. Then with a script created by us (Cleaner.py) we cleaned those data. We removed all words that contained a number, a "_" or that were composed of a single character.

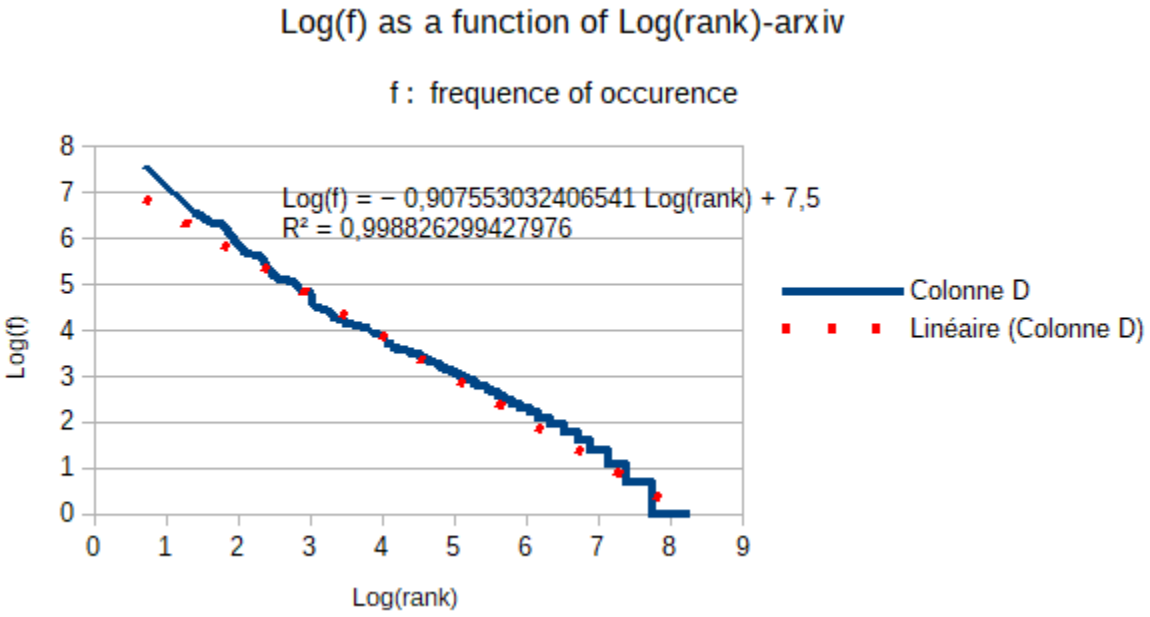
For each we plotted Log(freq) according to Log(rank) because we knew that a linear relationship connects these two quantities. From these linear relationships we were able to find the constants of the Zipf's law stated above. Let us now look more precisely at how we proceeded.

Analysis & Conclusion

Here are the graphs for each index :







Equations displayed on graphs are that of the linear regression curve. We notice that the coefficient R^2 it comes close to 1 for each. We can therefore say that a linear relationship links the two variables. Now let see how the values of this equation give us the constants of Zipf's law.

$$Log(f) = b_1 + a_1 Log(rank) \Leftrightarrow e^{Log(f)} = e^{b_1 + a_1 . Log(rank)} \Leftrightarrow f = e^{b_1} \times rank^{a_1}$$

Zipf's law : $f = \frac{c}{(rank+b)^a}$ therefore by identification $c = e^{b_1}$, $b = 0$ and $a = -a_1$

In order to get the best value, we have fixed the value of the ordinate at the origin ourselves so that the linear regression curve corresponds best with the straightest part of the initial curve because the extremes of the curve are noisy.

The best triplet for each index

index	a	b	c
20newsGroup	1,036	0	16317,6071980154
novels	1,263	0	1202604,28416478
arixv	0,907	0	1808,04241445606

We expected it to be the novels index that got the best values but given the graphs for it is 20newsGroup that has the best values because its representation of Log(f) as a function of Log(rank) is the one we managed to best match with a linear relationship.

Heap's law

Summary

The purpose of this second part was to find the Heap's law. We recall that the Heap's law allows to obtain the number ***d*** of different words in a text of which contains a number ***N*** of words thanks to the following relationship :

$d = K \times N^{\beta}$

For this we had at our disposal a set of texts called "novels". This set consists of 33 texts from novels. From this set we created 5 indexes with ElasticSearch. We know the size in terms of words of each index because before creating an index from texts of the set "novels" we took to look at the size of each of these texts in word processing software.

As a resource for this lab we had a python script called CountWords, which was making us able to count the number of different words in an ElasticSearch Index. We were able to know the size in terms of words ***N*** of an index and its number of different words ***d***.

To find the Heap's law with these data we chose to draw the curve that gives ***Log(d)*** as a function of ***Log(N),*** because in our lesson we saw that a linear relationship links these values with this relationship :

$$Log(d) = k_1 + \beta\, Log(N)$$

We will now explain more precisely how from this process we found the parameters of the law of heaps stated more and also how we chose the data on which we worked

Data

We have chosen to create 5 indexes in order to have enough points for our chart and therefore be relevant.

For the size of the indexes we wanted it to gradually increase from one index to another in order to have a good distribution of points on our chart.

Indexes have no text in common because we thought that it could distort the number of different words from one index to another if they were made up of identical texts :

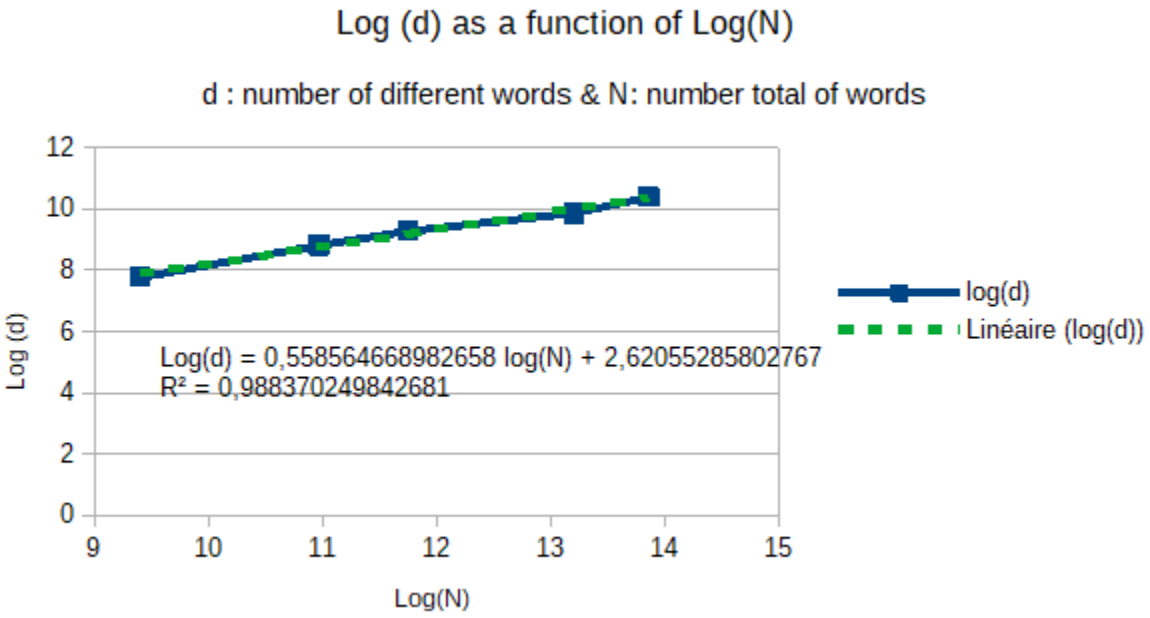
- The first index : 12 110 words, 2 texts
- The second index : 58 340 words , 1 text
- The third index : 126 736 words, 3 texts
- The fourth index : 544 079 words, 2 texts
- The fifth index : 1 050 047 words, 12 texts

After running the CountWords script on each index, we got the following data :

index	d	N	log(N)	log(d)
1	2 417	12 110	9,40178683654714	7,7902823807034
2	6 826	58 340	10,9740432434217	8,8284941294666
3	10 868	126 736	11,7498614617041	9,2935779705462
4	19 052	544 079	13,206849735882	9,8549273619207
5	32 885	1 050 047	13,864345483036	10,4007719057355

Analysis & Conclusion

With our previous results we were able to draw this graph :



The equation displayed on the graph is that of the linear regression curve. We notice that the coefficient R^2 it comes close to 1. We can therefore say that a linear relationship links the two variables. Now let see how the values of this equation give us the constants of Heap's law.

$$Log(d) = k_1 + \beta_1 Log(N) \Leftrightarrow e^{Log(d)} = e^{k_1 + \beta_1 . Log(N)} \Leftrightarrow d = e^{k_1} \times N^{\beta_1}$$

Heap's law : $d = K \times N^{\beta}$ therefore by identification $K = e^{k_1}$ and $\beta = \beta_1$

Rounding β_1 to 0,56 and k_1 to 2,62 so our Heap's Law is $d = e^{2,62} \times N^{0,56}$.

In order to see the quality of the coefficients we found we calculated the relative difference between the actual number of different words in each index and that obtained with the Heap's law with our coefficients :

N	$d = e^{2,62} \times N^{0,56}$	difference between the theoretical value and the recorded value in %
12 110	2657,12669597197	9,03708115747697
58 340	6409,036434063	6,50586980159221
126 736	9896,3612549470	9,81814143624992
54 4079	22378,050120660	14,8630023738744

1 050 047	32339,137742023	1,687930773948

We notice that this difference is around 10% for most values which is not acceptable. However for the index which has a size of more than one million words the gap between reality and calculation is less than 2% which is very satisfying.It would therefore have been interesting to test our law, with its coefficient obtained with our method, on texts of the size of the order of a million to see if our law is more reliable the larger the text size is.