The Most Frequent Words

António Ramos, ajframos@ua.pt, 101193, MEI

***Resumo* – Este trabalho apresenta uma análise comparativa dos contadores exatos e space-saving count**
Para o estudo, foi usado livros de grandes dimensões de língua portuguesa, inglesa e espanhola.

***Abstract* - This work presents a computational comparation of exact counter and space-saving count**
For the study, has used large books of Portuguese language, English, French and Spanish.

I. Introduction

In nowadays exists then need to count the most frequent words, in this work is analysed the exact counts and the algorithm from frequent words called space saving count.

For this comparison, it's used a book in English "The Bible", a book in Portuguese "Os Maias -Eps da Vida Romântica" and a book in Spanish "Don Quijote". And it's counted each word of the books.

II. Algorithm analysis

In this chapter is it's showed how the algorithm was analysed and what is the approach chosen.

A. Handle information

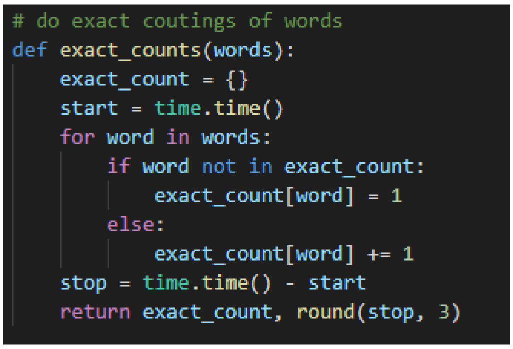
Due to free choice on book, I choose "The Bible" to represent English, "Os Maias -Eps da Vida Romântica" to represent Portuguese and a "Don Quijote" to represent Spanish.

Since the requisite for this work is to remove the stop words from the counting of words, it's used the package of python nltk, that contains a list of stop words of all languages and it's also had a tokenizer for the text. It's need it to install the package to the program works.

The choice of the book is made on execution of the main program. So, it's processed one book at time.

B. Exact Counter

To do exact counter is created a dictionary that saves the word in lowercase and the counting's. (Figure 1).

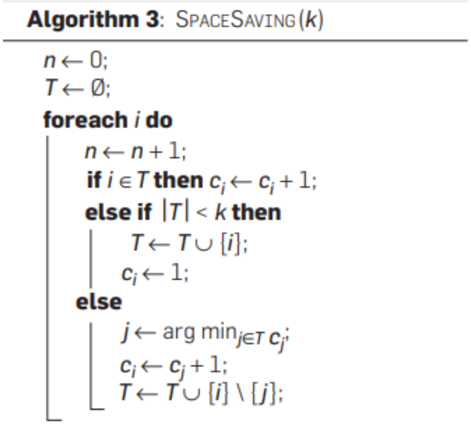


```python
# do exact coutings of words
def exact_counts(words):
    exact_count = {}
    start = time.time()
    for word in words:
        if word not in exact_count:
            exact_count[word] = 1
        else:
            exact_count[word] += 1
    stop = time.time() - start
    return exact_count, round(stop, 3)
```

Figure 1 – Exact count

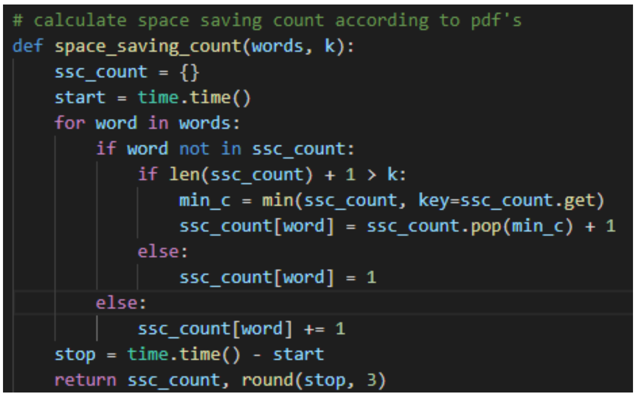
C. Space-Saving-Count

The algorithm of space saving count is implemented according to the slides and the author [1] (figure 2). It receives a list of all words of a document and a k, that's chosen manually, the values are 10, 25, 50 and 70. It saves the counting's of each word in a dictionary. For counting he checks if a word is on the dictionary and if not then verifies if the length of dictionary + 1 is bigger than the value stipulated then calculate the min counter of all words that is on dictionary and then removes it and add one count to the dictionary of that word. If the word exists counts more one value.



```
Algorithm 3: SpaceSaving(k)
n ← 0;
T ← ∅;
foreach i do
    n ← n + 1;
    if i ∈ T then c_i ← c_i + 1;
    else if |T| < k then
        T ← T ∪ {i};
        c_i ← 1;
    else
        j ← arg min_{j∈T} c_j;
        c_i ← c_j + 1;
        T ← T ∪ {i} \ {j};
```

Figure 2 – Space Saving Count of [1]



```python
# calculate space saving count according to pdf's
def space_saving_count(words, k):
    ssc_count = {}
    start = time.time()
    for word in words:
        if word not in ssc_count:
            if len(ssc_count) + 1 > k:
                min_c = min(ssc_count, key=ssc_count.get)
                ssc_count[word] = ssc_count.pop(min_c) + 1
            else:
                ssc_count[word] = 1
        else:
            ssc_count[word] += 1
    stop = time.time() - start
    return ssc_count, round(stop, 3)
```

Figure 3 – Space Saving Count algorithm

III. Tests Sequence Analysis

The tests performed are in the directories “results” and “results_rel_error”.

A. *Execution Time Analysis*

The analysis regarding the execution time is given in seconds is in the bellow image.

The image has exact count, space saving count 10, 25, 50, 70. It’s analysed for all books, shown above. The time of execution of the counter exact is smaller than space saving counter. In space saving count when the k grown, the time also grown. This information is written to a file called “*results_time.csv*” and to an image, for visuals proposals, called “*results_time.png*”, in each execution of program.

I. The Bible.

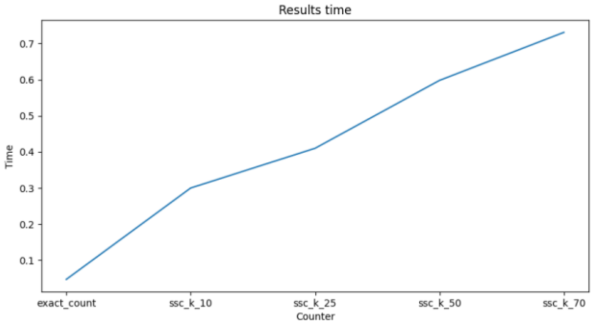


Figure 4 - Execution Time – The bible

II. Don Quixote.

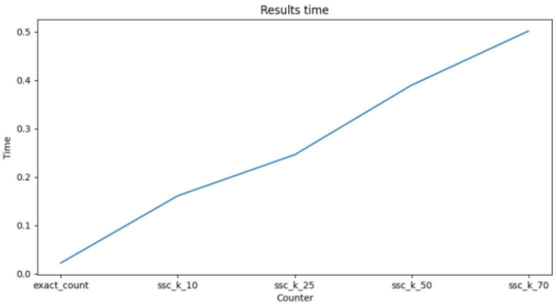


Figure 5 - Execution Time – Don Quixote

III. Os Maias

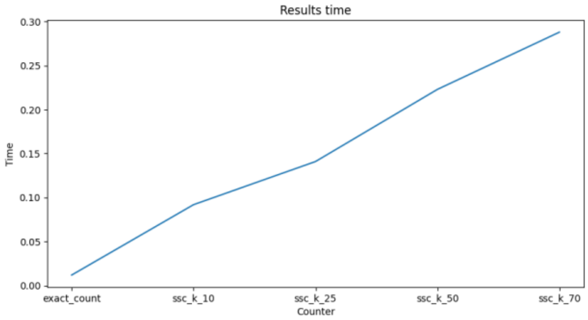


Figure 6 - Execution Time – Os Maias

B. *Relative Error*

To calculate the relative error for the algorithm space saving count, I based on expression of *[2]*
*|exact_counter – ssc_counter |/exact_counter * 100*
This information is written to a file called *“results_rel_error.csv”* in each execution of program. To visualize the information, it’s created a directory called “results_rel_error” that contains each k chosen. Since the k=10 is the k that haves least words, I will evaluate in each book. Regarding the values, it’s shown that the most frequent words have bigger error than the others.

I. The Bible

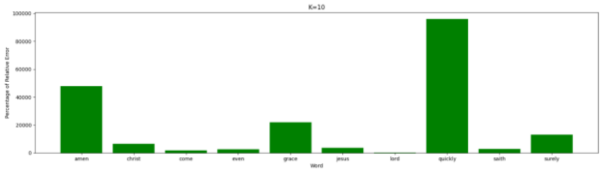
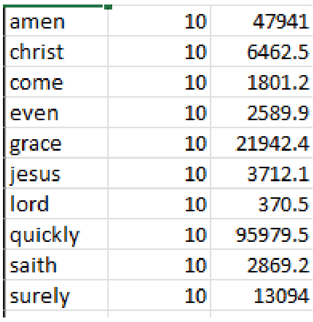


Figure 7 – Bar Chart Relative Error – The Bible



amen	10	47941
christ	10	6462.5
come	10	1801.2
even	10	2589.9
grace	10	21942.4
jesus	10	3712.1
lord	10	370.5
quickly	10	95979.5
saith	10	2869.2
surely	10	13094

Figure 8 – Values of Relative Error – The Bible

II. Don Quixote

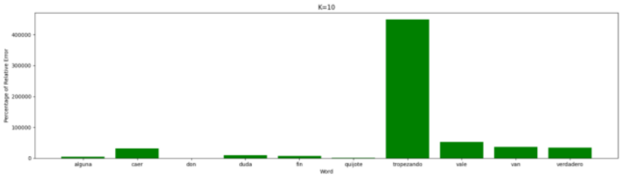
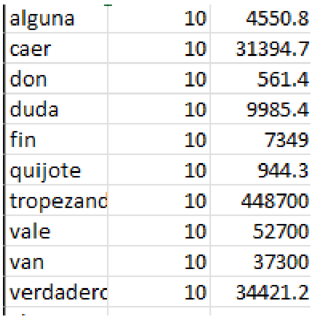


Figure 9 – Bar Chart Relative Error – Don Quixote



alguna	10	4550.8
caer	10	31394.7
don	10	561.4
duda	10	9985.4
fin	10	7349
quijote	10	944.3
tropezanc	10	448700
vale	10	52700
van	10	37300
verdaderc	10	34421.2

Figure 10 – Values of Relative Error – Don Quixote

III. OS Maias

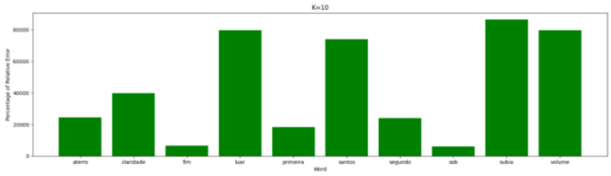
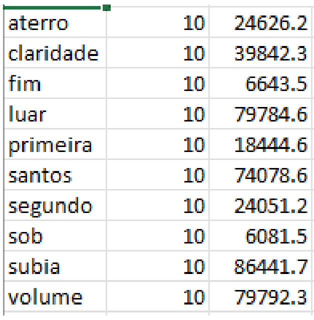


Figure 11 – Bar Chart Relative Error – Os Maias



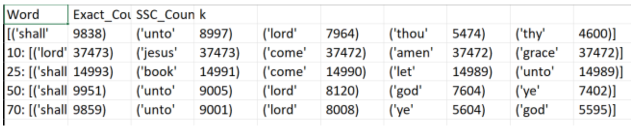
aterro	10	24626.2
claridade	10	39842.3
fim	10	6643.5
luar	10	79784.6
primeira	10	18444.6
santos	10	74078.6
segundo	10	24051.2
sob	10	6081.5
subia	10	86441.7
volume	10	79792.3

Figure 12 – Values of Relative Error – Os Maias

C. Most Frequent Item

Regarding the results for the most frequent item for exact count and space saving count, it has created a file "results_count_word". It's shown that if the k is bigger then he will be getting the counting's of exact count. For example, in k = 70 the counting's is almost equal than the exact count.

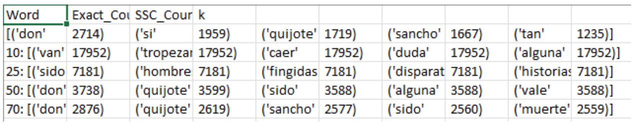
I. The Bible



Word	Exact_Cou	SSC_Coun	k						
[('shall'	9838)	('unto'	8997)	('lord'	7964)	('thou'	5474)	('thy'	4600)]
10: [('lord'	37473)	('jesus'	37473)	('come'	37472)	('amen'	37472)	('grace'	37472)]
25: [('shall	14993)	('book'	14991)	('come'	14990)	('let'	14989)	('unto'	14989)]
50: [('shall	9951)	('unto'	9005)	('lord'	8120)	('god'	7604)	('ye'	7402)]
70: [('shall	9859)	('unto'	9001)	('lord'	8008)	('ye'	5604)	('god'	5595)]

Figure 13 – Most Frequent Item – The Bible

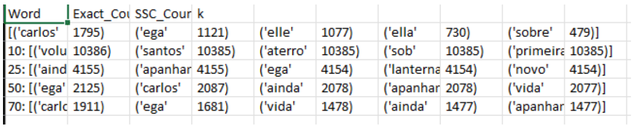
II. Don Quixote



Word	Exact_Cou	SSC_Cour	k						
[('don'	2714)	('si'	1959)	('quijote'	1719)	('sancho'	1667)	('tan'	1235)]
10: [('van'	17952)	('tropezar	17952)	('caer'	17952)	('duda'	17952)	('alguna'	17952)]
25: [('sido	7181)	('hombre:	7181)	('fingidas	7181)	('disparat	7181)	('historia:	7181)]
50: [('don'	3738)	('quijote'	3599)	('sido'	3588)	('alguna'	3588)	('vale'	3588)]
70: [('don'	2876)	('quijote'	2619)	('sancho'	2577)	('sido'	2560)	('muerte'	2559)]

Figure 14 – Most Frequent Item – Don Quixote

III. Os Maias



Word	Exact_Cou	SSC_Cour	k						
[('carlos'	1795)	('ega'	1121)	('elle'	1077)	('ella'	730)	('sobre'	479)]
10: [('volu	10386)	('santos'	10385)	('aterro'	10385)	('sob'	10385)	('primeira	10385)]
25: [('aind	4155)	('apanhar	4155)	('ega'	4154)	('lanterna	4154)	('novo'	4154)]
50: [('ega'	2125)	('carlos'	2087)	('ainda'	2078)	('apanhar	2078)	('vida'	2077)]
70: [('carlc	1911)	('ega'	1681)	('vida'	1478)	('ainda'	1477)	('apanhar	1477)]

Figure 15 – Most Frequent Item – Os Maias

IV. Conclusion

The relative error and the most frequent item indicate that bigger k has low relative error and it almost get right the counting's of exact count. So, the best k for these books is k = 70, even the time of execution is higher than exact count.

References

[1] D. Cono D'elia, "Mining heavy hitters with Space-Saving," 2013, Accessed: Jan. 24, 2022. [Online]. Available: http://developer.gnome.org/glib/.

[2] https://www.greelane.com/pt/ci%c3%aancia-tecnologia-matem%c3%a1tica/ci%c3%aancia/how-to-calculate-percent-error-609584/