A Theoretical Model for *n*-gram Distribution in Big Data *Corpora*

Joaquim F. Silva[1], Carlos Goncalves[1,2], Jose C. Cunha[1]
(1) NOVA Laboratory for Computer Science and informatics
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
2829-516 Caparica, Portugal
(2) ISEL — Instituto Superior de Engenharia de Lisboa, IPL — Instituto Politécnico de Lisboa
1959-007 Lisboa, Portugal
jfs@fct.unl.pt, cgoncalves@deetc.isel.pt, jcc@fct.unl.pt

***Abstract*—There is a wide diversity of applications relying on the identification of the sequences of *n* consecutive words (*n*-grams) occurring in *corpora*. Many studies follow an empirical approach for determining the statistical distribution of the *n*-grams but are usually constrained by the *corpora* sizes, which for practical reasons stay far away from Big Data. However, Big Data sizes imply hidden behaviors to the applications, such as extraction of relevant information from Web scale sources.**

In this paper we propose a theoretical approach for estimating the number of distinct *n*-grams in each *corpus*. It is based on the Zipf-Mandelbrot Law and the Poisson distribution, and it allows an efficient estimation of the number of distinct 1-grams, 2-grams,. . . , 6-grams, for any *corpus* size. The proposed model was validated for English and French *corpora*. We illustrate a practical application of this approach to the extraction of relevant expressions from natural language *corpora*, and predict its asymptotic behaviour for increasingly large sizes.

***Keywords*-*n*-gram Models; Big Data; Zipf-Mandelbrot Law; Poisson Distribution; Extraction of Relevant Expressions**

I. Introduction

Words do not occur with similar probabilities in text. Everyday experience shows us that words such as ”the”, ”and”, or ”in” are much more frequent than ”agriculture” or ”stomatology”, whatever the topic of the text. This means words are more or less repeated throughout the text, so the number of distinct words in a *corpus* is less than the total number of words in that *corpus*. This applies to single words or multiwords (sequences of n consecutive words where $n \geq 2$). Most of the empirical studies on the *n*-gram distribution only cover *corpora* of relatively small sizes. This precludes their usage towards understanding the behaviour of an increasingly large number of Big Data applications that depend on the *n*-gram distributions. This requires an accurate estimation of the repetition patterns of the n-grams and their evolution for increasing-size *corpora*.

Zipf Law [1] states that the frequency[1] of any word in a work of literature is inversely proportional to its rank in the frequency table. Mandelbrot [2] proposed a generalization of Zipf Law showing to be a more adequate model, but is not sufficient to estimate the number of distinct n-grams.

We propose an efficient approach to estimate this number, for single or multiwords, for any *corpus* size. It is based on the properties of the Poisson distribution and the Zipf-Mandelbrot Law. Results for English and French are shown. In this paper we discuss related work (Sect. II), followed by the proposed approach (Sect. III), results and applications (Sect. IV) and conclusions (Sect. V).

II. Related Work

The frequency distribution of words in text has been studied in statistical linguistics ([1], [2], [3]). These frequencies tend to follow the Zipf Law [1], [4]. However, Mandelbrot [2] proposed a generalization of this law for a better fitting of the frequency of some ranks, as it happens in some *corpora*. Other improvements to Zipf Law were proposed in [5]. A critical review about Zipf’s word frequency law in natural language is made in [6] claiming that semantics strongly influences word frequency. We think that is true, but it will not be the case for very Big Data *corpora*, where the existence of numerous topics tends not to favor any particular topic.

Heaps’ law, originally discovered by Gustav Herdan [7], is an empirical law describing the number of distinct single words in documents as a function of the document length. It states that $V_R(n) = K\, n^{\beta}$ where V_R is the number of distinct words in the text of size n, and K and β are parameters determined empirically. According to [8], [9], [10], under mild assumptions, this law is asymptotically equivalent to Zipf Law concerning the frequencies of individual words.

In [11], [12], although their aim is not to propose an approach to estimate the number of distinct n-grams, some estimate of cardinality is presented for hashing design. However, these estimates incur some computational weight, depending on the data volume.

A clustering model for words distribution [13] is proposed based on estimating a probability (p) for each word occurring in a document of a given length. Then, the number of distinct words can be estimated but the model is not very accurate for

[1] In this paper, by default, the term ”frequency” refers to the absolute number of occurrences.

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

large p values, and is not always a close fit to observed data. There is no evidence that this approach could be extended to larger n-gram sizes: 2-grams, 3-grams, $\ldots$.

To the best of our knowledge, there is no approach focused on the estimation of the number of distinct multiword n-grams. In this paper a new approach is proposed to estimate the cardinality of n-grams (single or multiwords) in English or other languages.

III. The Number of Distinct n-grams in *Corpora*

This section presents an approach to estimate the cardinality of distinct n-grams for any *corpus* size.

A. The Zipfian Models

The most widely recognized approach to model the distribution of words in text is the Zipf Law [1]. It states that the frequency of the r^{th} most frequent word in natural *corpora*, $f(r)$, scales according to

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{1}$$

where r is the frequency rank of a word, and $\alpha \approx 1$. The most frequent word corresponds to $r = 1$; for the i^{th} most frequent word ($r = i$), its frequency $f(i)$ is proportional to $1/i^{\alpha}$. Though Zipf Law works as a good model, it presents some deviations mainly for low and high ranks for some *corpora*. To minimize these deviations, Mandelbrot [2] proposed a generalization of this law by $shifting$ rank r by a value β:

$$f(r) \propto \frac{1}{(r+\beta)^{\alpha}} \quad . \tag{2}$$

Then we state a corresponding equation for the frequency of rank r, denoted by $f(r,(\beta,\alpha))$ for $r = 1, 2, \ldots$, keeping the proportionality according to the Mandelbrot model in (2):

$$f(r,(\beta,\alpha)) = (1+\beta)^{\alpha} \, \frac{f(1)}{(r+\beta)^{\alpha}} \quad , \tag{3}$$

where $f(1)$ means the frequency of the most frequently occurring word in *corpus*. The particular case of $\beta = 0$ in (3) corresponds to the Zipf Law model.

By using (3), (β, α) combinations can be searched for each *corpus* so that the estimate of the frequency of each rank is as close as possible to the actual frequency in *corpus*. As shown in Sect. IV and resulting from our experiments, we consider that for each language and each n-gram size, there is a *best* (β, α) *combination*, which produces the most possible accurate results, similar to different *corpus* sizes, to estimate the number of distinct n-grams.

B. Estimating the Number of Distinct Single Words in Corpora

Having analyzed the relative frequency of the most common word in English, that is ”the”, corresponding to $r = 1$, — it can be taken as the probability of rank 1 in the *corpus*, denoted by p_1 — we noticed that it tends to be constant when *corpora* sizes grow, as this probability presents very slight variations for small and median size *corpora*, but converging to a value keeping more and more decimal digits for large *corpora*: for English *corpora* over 500 million words, the first 7 decimal digits showed to be the same for p_1, which was 0.0503705. Thus, considering that the semantics of the texts is kept if words are separated from some punctuation marks, then, in order to obtain a more correct counting for the actual number of occurrences of each word, words were separated from the following set of characters, for all *corpora* in this paper: {’<’, ’>’, ’”’, ’!’, ’?’, ’:’, ’;’, ’,’, ’(’, ’)’, ’[’, ’]’, ’.’}. If other criteria are used for the content of this set, p_1 can tend to a different constant value.

For other ranks of very frequent English words such as ”of”, ”and”, ”in”, among others, it was easily verified that their individual probabilities also tend to constant values. We noticed this tendency still exists for other not so frequent words such as ”late” and ”again”, although larger *corpora* were needed to reach their corresponding constant probabilities. Thus, there is no reason not to believe that this tendency is applicable to all words, even for rare ones, which will certainly be verified in huge *corpora*. For the other language considered in this work (French), the same behaviour was verified. This leads us to the belief that as *corpora* sizes grow for the same language, words tend to have fixed ranks, which is consistent with the existence of what we called the *best* (β, α) *combination* for each language.

Thus, assuming that the probability of rank 1 tends to remain constant for large *corpora*, then its expected frequency is $f(1) = p_1 \times c$, where c is the *corpus* size in words. Also, the frequency of rank r can be estimated by (3), for a given (β, α) combination, such that $f(r,(\beta,\alpha)) = (1+\beta)^{\alpha} f(1)/(r+\beta)^{\alpha}$. So, the expected frequency of rank r in a *corpus* having c words, for a (β, α) *combination*, is:

$$f(r,(\beta,\alpha),c) = (1+\beta)^{\alpha} \, \frac{p_1 \times c}{(r+\beta)^{\alpha}} \quad . \tag{4}$$

However, once there is a *best* (β, α) *combination* for each language and that *best combination* must be used to obtain the frequency of rank r in a *corpus* of size c, in a language l, then β and α depend on l. Similarly, the probability of rank 1 depends on language l too:

$$f(r,l,c) = (1+\beta(l))^{\alpha(l)} \, \frac{p_1(l) \times c}{(r+\beta(l))^{\alpha(l)}} \quad . \tag{5}$$

Considering Poisson Distribution: A random variable X follows a Poisson distribution [14] with parameter $\lambda > 0$, if, for $k = 0, 1, 2, \ldots$, the probability mass function is, according to the classical notation:

$$f(k;\lambda) = Pr(X = k) = \frac{\lambda^{k} e^{-\lambda}}{k!} \tag{6}$$

where e is the Euler's constant ($e = 2.71828\ldots$) and $k!$ is the factorial of k. λ is a positive real number equal to the expected value of X. This distribution provides a realistic model for many random phenomena for which a count of some sort is of interest, such as the number of traffic accidents per week, given its rate λ. So, let X be the number of times that word w in rank r occurs in a *corpus* of size c, written in language l. Then, the probability of non-occurrence of w is:

$$Pr(X = 0) = \frac{\lambda^0 e^{-\lambda}}{0!} = e^{-\lambda} = e^{-f(r,l,c)} \tag{7}$$

where $f(r,l,c)$ stands for the expected frequency of rank r, given by (5). Thus, the probability of w occurring in the same *corpus* is:

$$Pr(X \geq 1) = 1 - e^{-f(r,l,c)} \quad . \tag{8}$$

Considering for example, the size of the smallest English *corpus* used in this paper (c = 2,226,162, l =”English”) and using the *best* (β, α) *combination* searched for English 1-grams ($\beta = 7.7950$, $\alpha = 1.3466$) — explained in Sect. IV —, by application of (8), the probabilities of occurrence of the words corresponding, for example, to ranks 1, 100,000 and 3,000,000, are $Pr(X \geq 1) = 1.0$, $Pr(X \geq 1) = 0.32120$ and $Pr(X \geq 1) = 0.00396$ respectively. For a *corpus* 10 times larger, those probabilities are respectively $Pr(X \geq 1) = 1.0$, $Pr(X \geq 1) = 0.97923$ and $Pr(X \geq 1) = 0.03895$. This matches our intuition, as we expect that the probability of occurrence of frequent words is high, even in small *corpora*; and the probability of rare words (higher ranks, such as 3,000,000), though low, grows with the *corpus* size.

So, by (8) it is possible to calculate the probability of any word (its rank) in the vocabulary of a language. Now, in order to calculate the number of distinct words in a *corpus* with c words written in a language l, we propose to sum the probabilities of occurrence of each word in that *corpus*, considering the entire vocabulary of l, that is its size $\mathcal{V}(l)$:

$$\begin{aligned} Dist(l,c) &= \sum_{r=1}^{r=\|\mathcal{V}(l)\|} \left(1 - e^{-f(r,l,c)}\right) \\ Dist(l,c) &= \|\mathcal{V}(l)\| - \sum_{r=1}^{r=\|\mathcal{V}(l)\|} e^{-\left((1+\beta(l))^{\alpha(l)} \times \frac{p_1(l)\times c}{(r+\beta(l))^{\alpha(l)}}\right)} \end{aligned} \tag{9}$$

where $\|\mathcal{V}(l)\|$ is the size of $\mathcal{V}(l)$, being the number of distinct words of l, which can reach some millions for languages such as English or French, as we noticed for large *corpora*.

This sum of probabilities must not be taken as a probability. Indeed, the sum of probabilities can be used to estimate some population sizes. For example, the number of heads from tossing a fair coin 1000 times can be correctly estimated by summing the probability of a head in a single toss (0.5), 1000 times, i.e. $\sum_{t=1}^{t=1000} 0.5 = 500$.

C. Estimating the Number of Distinct Multiwords in Corpora

Multiwords, also known as n-grams ($n \geq 2$) are occurrences of contiguous words in text. ”in the”, ”United Nations”, are 2-grams; ”in this world”, ”let it be” are 3-grams; etc.. We noticed that the frequency of 2-grams in *corpora* also follows a Zipfian distribution. For English *corpora*, the most common 2-gram (i.e. $r = 1$), ”of the”, also tends to have a relative frequency that converges to a fixed value when *corpora* grow. The same happens to ranks $2, 3, \ldots$. So, the estimate of frequency given by (5) could also be applied to 2-grams, however, there is a different *best* (β, α) *combination* for 2-grams for each of the considered languages. Likewise for larger size n-grams: 3-grams, 4-grams,$\ldots$. Thus, due to their dependence on the language l and on the n-gram size, β and α are denoted by $\beta(l,n)$ and $\alpha(l,n)$. Similarly, the probability of rank 1 also depends on the n-gram size and on each specific language, being denoted by $p_1(l,n)$.

On the other hand, for the same language, the number of distinct single words is different from the number of distinct 2-grams, 3-grams,$\ldots$. This means that the size of the vocabulary depends not only on the language, but also on the n-gram size; we use $\|\mathcal{V}(l,n)\|$ to denote this number.

Therefore, (9) can be generalized also to any n-gram size:

$$Dist(l,n,c) = V - \sum_{r=1}^{r=V} e^{-\left((1+\beta(l,n))^{\alpha(l,n)} \times \frac{p_1(l,n)\times c}{(r+\beta(l,n))^{\alpha(l,n)}}\right)} \tag{10}$$

where symbol V means $\|\mathcal{V}(l,n)\|$.

D. An Efficient Implementation

Using (10) we can estimate the number of distinct n-grams in a language for each *corpus* size. However this is computationally heavy; e.g. for the case of 6-grams for any English *corpus*, due to the large vocabulary size, the sum in (10) may lead to an order of magnitude of 10^{11} iterations. So, we propose an efficient implementation of (10), where the heavily iterated sums are replaced with an integral, leading to (17), which is derived as follows.

According to the Euler-Maclaurin formula, the finite sum $\sum_{n=a}^{n=b} g(n)$ can be substituted by an integral as follows [15]:

$$\sum_{n=a}^{n=b} g(n) \approx \int_a^b g(x)dx + B \tag{11}$$

where $B = \frac{g(b)+g(a)}{2} + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!}\left(g^{(2k-1)}(b) - g^{(2k-1)}(a)\right)$ and $g^{(2k-1)}(b)$ stands for the $(2k-1)^{\text{th}}$ derivative of g(.) in b. And B_m, a Bernoulli number, is given by:

$$B_m = \sum_{v=0}^{v=m} \sum_{j=0}^{j=v} (-1)^j \binom{v}{j} \frac{j^m}{v+1} \quad . \tag{12}$$

So, by (12), $B_2 = 1/6$, $B_4 = -1/30$, etc..

On the other hand, variables l, n, c, V, $\beta(l,n)$ and $\alpha(l,n)$ in (10), can be taken as constants in the context of each specific $Dist(l,n,c)$ calculation, that is, in the context of the estimation of the number of distinct n-grams of size n

of a *corpus* size c in a specific language l. Thus, (10) can be simplified for better mathematical manipulation, as follows:

$$\begin{aligned}
&D = Dist(l,n,c) \quad A = \alpha(l,n) \quad R = r + \beta(l,n)\\
&Q = (1+\beta(l,n))^{\alpha(l,n)} \times p_1(l,n) \times c\\
&r_1^\star = 1 + \beta(l,n) \quad r_v^\star = \|\mathcal{V}(l,n)\| + \beta(l,n) \ .
\end{aligned} \tag{13}$$

Thus

$$D = V - \sum_{R=r_1^\star}^{R=r_v^\star} e^{-Q\,R^{-A}} \ . \tag{14}$$

Then, by applying the Euler-Maclaurin formula, given by (11), taking R as the integration variable,

$$D = V - \sum_{R=r_1^\star}^{R=r_v^\star} e^{-Q\,R^{-A}} = V - \left(\int_{r_1^\star}^{r_v^\star} e^{-Q\,R^{-A}} dR + \ B \right) \tag{15}$$

where

$$B = \frac{g(r_1^\star)+g(r_v^\star)}{2} + \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} \big(g^{(2k-1)}(r_v^\star) - g^{(2k-1)}(r_1^\star)\big), \tag{16}$$

$g(R) = e^{-Q\,R^{-A}}$ and B_{2k} is given by (12).

Our experiments showed us that, it made a negligible difference to the result of D in (15), including or not the derivatives of $g(.)$. in the sum of (16). So, for simplicity, their respective equations are ignored in this paper.

Then, following the rule of the *integration by substitution*, $\int_a^b g(x)dx$ is equal to $\int_{\varphi^{-1}(a)}^{\varphi^{-1}(b)} g(\varphi(t))\,\varphi'(t)dt$. So, let $t = R^{-A}$ and $R = \varphi(t) = t^{-1/A}$, and then

$$\begin{aligned}
&D - V + B = -\int_{r_1^\star}^{r_v^\star} e^{-Q\,R^{-A}} dR\\
&= -\int_{\varphi^{-1}(r_1^\star)}^{\varphi^{-1}(r_v^\star)} e^{-Q\,t}\,(t^{-\frac{1}{A}})' dt = \frac{1}{A}\int_{\varphi^{-1}(r_1^\star)}^{\varphi^{-1}(r_v^\star)} e^{-Q\,t}\,t^{(\frac{-1}{A}-1)} dt\\
&= \frac{1}{A}\left[-\frac{\Gamma\left(-\frac{1}{A}, Q\,t\right)\,(Q\,t)^{\frac{1}{A}}}{t^{\frac{1}{A}}} + Const \right]_{t=\varphi^{-1}(r_1^\star)}^{t=\varphi^{-1}(r_v^\star)}\\
&= -\frac{Q^{\frac{1}{A}}}{A}\left[\Gamma\left(-\frac{1}{A}, Q\,t\right) + Const \right]_{t=\varphi^{-1}(r_1^\star)}^{t=\varphi^{-1}(r_v^\star)} \ .
\end{aligned}$$

Then,

$$D = E - \frac{Q^{\frac{1}{A}}}{A}\left[\Gamma\left(-\frac{1}{A}, Q \times (r_v^\star)^{-A}\right) - \Gamma\left(-\frac{1}{A}, Q \times (r_1^\star)^{-A}\right) \right] \tag{17}$$

where $E = V - B$ and $\Gamma(.,.)$ is the Incomplete Gamma function. Thus, (17) gives the number of distinct n-grams, where the substitutions for symbols D, Q, A, $r_v^\star$ and $r_1^\star$ are in (13), $V = \|\mathcal{V}(l,n)\|$, $g(R) = e^{-Q\,R^{-A}}$ and B in (16).

For testing the efficiency of this approach, we used a laptop with Mac OS X 10.5.8, 2.4 Ghz Intel, 4Gb 667 MHz DDR2 SDRAM. When (10) was used to estimate the number of distinct 1-grams of a 100,000,000 words English *corpus*, it took 178.73 minutes. The same estimate took 0.0078408 seconds by using (17). Similar gains were obtained for larger n-gram sizes. The vocabulary size and the β and α values used in these tests result from the tuning phase (Sect. IV-A). Both implementations were written in Python 2.5.1.

IV. Results

This approach was tested with English and French Wikipedia based *corpora* [16]. To assess the accuracy of the estimates as the *corpus* size grows, for each language, *corpora* were generated by doubling approximately the size of each *corpus*, from about 2×10^6 to 10^9 words: 2×10^6, 4×10^6, 8×10^6, $\ldots$. For obtaining each of these specific *corpora* sizes, random paragraphs were extracted from the largest *corpus* (10^9 words) in each language until the required size is approximately reached. We identify each *corpus* by its size in words[2]. Tables I and II show, for each *corpus*, its size and the number of actual distinct n-grams.

A. Tuning Parameters for each Language

In order to obtain the best possible estimate of the number of distinct n-grams for a given *corpus* by using (17), three values must be previously found: $\beta(l,n)$, $\alpha(l,n)$ and $\|\mathcal{V}(l,n)\|$, corresponding to the *best* (β, α) *combination* and the vocabulary size for each pair (language, n-gram size).

These tunings were made according to the following criterion. For each pair, an exhaustive search was made varying $\|\mathcal{V}(l,n)\|$ from 2×10^7 up to a maximum of 4×10^{11} words, by steps of 1×10^6 words or larger; then, for each $\|\mathcal{V}(l,n)\|$ value, different (β, α) combinations were taken by varying α from 0.5 to 1.8 and β from -0.5 to 80, by steps of 0.005 and 0.0001 respectively. Then, for each ($\|\mathcal{V}(l,n)\|$, β, α) combination, two estimates were obtained using (17): one for a relatively small *corpus* and another one for a relatively large *corpus*. Next, if these two estimates did not deviate more than 5% from the actual number of distinct n-grams of the respective *corpus*, the search stopped as we considered that the actual size of the corresponding vocabulary could be approximated by ($\|\mathcal{V}(l,n)\|$) and the *best* (β, α) *combination* had been found. Although actual vocabularies are open, as new words and multiwords arise and others tend to stop being used, they are finite. However, the lack of consensus about the real vocabulary sizes, prevent us from assessing how far the $\|\mathcal{V}(l,n)\|$ values are from the actual sizes.

Tables III and IV show the *best* (β, α) *combination* and vocabulary size for each pair (language, n-gram size), resulting from each tuning. Then, the obtained parameter values were used in (17) to estimate the number of distinct n-grams for all *corpora* of Tables I and II. The relative errors of these estimates are shown in Tables V and VI. This error

[2] All these *corpora* are available at http://cjsg.dynip.sapo.pt/corpus-demos/BigData2016/

Table I
THE ACTUAL NUMBER OF DISTINCT n-GRAMS FOR EACH ENGLISH *corpus*

***Corpus* Size**	**1-grams**	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
2,226,162	171,011	918,150	1,682,247	2,031,928	2,146,452	2,188,946
4,450,249	275,142	1,604,333	3,168,994	3,971,242	4,253,952	4,358,108
8,955,079	446,746	2,797,067	5,961,183	7,775,790	8,465,778	8,721,741
18,006,731	728,634	4,833,505	11,104,396	15,128,835	16,789,872	17,421,851
35,771,592	1,186,891	8,207,918	20,257,703	28,895,723	32,773,139	34,304,469
72,677,601	1,966,084	14,086,371	37,403,872	56,034,675	65,160,115	68,934,903
140,275,807	3,155,397	23,084,447	65,483,074	102,858,205	122,712,271	131,338,738
245,492,006	4,718,348	34,960,884	104,706,565	171,430,164	209,426,765	226,713,292
490,846,877	7,783,551	57,967,910	185,346,762	319,964,031	403,573,252	444,167,811
981,996,022	12,813,557	94,705,122	323,192,231	589,842,301	770,074,139	863,071,391

Table II
THE ACTUAL NUMBER OF DISTINCT n-GRAMS FOR EACH FRENCH *corpus*

***Corpus* Size**	**1-grams**	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
2,172,301	157,116	766,994	1,509,595	1,904,418	2,050,038	2,107,627
4,322,189	248,002	1,301,878	2,770,059	3,653,037	4,008,093	4,151,262
8,710,051	394,498	2,209,397	5,087,526	7,041,834	7,899,233	8,256,115
17,442,724	628,094	3,713,547	9,205,032	13,382,741	15,384,969	16,252,632
34,744,534	996,291	6,160,934	16,420,573	25,131,306	29,684,624	31,749,122
69,331,062	1,582,633	10,162,503	29,033,342	46,814,925	56,956,887	61,782,559
139,025,258	2,517,866	16,682,341	50,978,036	86,678,234	108,858,991	119,959,588
242,346,014	3,654,954	24,662,841	79,264,438	140,574,704	181,301,336	202,674,252
484,314,987	5,778,693	39,559,507	135,285,205	253,300,084	338,323,948	385,396,971
970,351,308	9,254,004	63,154,520	228,935,214	451,752,256	625,662,563	726,463,547

is given by $(Es/Act - 1) \times 100\%$, where Es and Act stand for the estimate and the corresponding actual number in the *corpus*, respectively. Results show that the relative error is generally less than 1% for estimates of 2-grams,…, 5-grams for the full range of *corpora* in both languages. However, errors are slightly higher for some of the *corpora*, reaching a maximum of 4.3% for 1-grams, and 4.7% for 6-grams.

Fig. 1 illustrates the evolution of the actual numbers of distinct 1-grams and 2-grams, and their respective estimates obtained by this approach for the English *corpora* referred in the Tables I and V. It shows a small deviation between the curves of 1-grams, however not more than 4.3% as we know from table V. For 2-grams, curves coincide apparently, since they deviate less than 1% for all *corpus* sizes. Table V allows us to preview similar coincidence for 3-grams, 4-grams and 5-grams; curves for 6-grams would also present just a small deviation, as in the 1-gram case. Similar curves were obtained for French using Table VI.

B. Estimates for Big Data Corpora

Estimates are given by (17), an efficient implementation of (10). From (10) we conclude that, for each n-gram size, as the *corpus* size increases towards infinity, the sum in the second parcel tends to zero, so the number of distinct n-grams tends to the size of the vocabulary. Due to this, the evolution of the number of distinct n-grams as a function of the *corpus* size exhibits a *plateau* which corresponds to the respective vocabulary size. This is illustrated in Fig. 2 whose left curve shows that for English *corpora* sizes larger than a *threshold* of 9.22×10^{11} words, the estimated number of distinct 1-grams will not grow further, having reached the vocabulary size, that is 1.95×10^8 words. The right curve shows the corresponding values for the 6-grams case. The values for the French language are shown in Fig. 3.

Tables VII and VIII show all obtained *plateau* values for the different n-gram sizes and the corresponding *corpus* size thresholds from which these *plateau* values are reached.

C. Applications

The estimation of the number of distinct n-grams in different size *corpora* for Big Data is critical to support algorithm design and implementation. The identification of the *plateau* levels allows to determine the maximum required capacity of memory and number of machines (for distributed implementations) in applications whose problem size is proportional to the number of distinct n-grams, *e.g.* in the LocalMaxs method [17], which counts n-gram frequencies, calculates n-gram internal cohesions, and extracts relevant n-grams. A cache for n-grams was designed to keep the distinct n-grams locally, taking advantage of their repetition patterns, with a significant reduction in the total execution time [18]. According to the model, the cache miss ratio tends to constant values for Big Data *corpora*, as determined by the *plateaux* defined in this paper. These conclusions also apply to similar applications.

V. CONCLUSIONS

In this paper we propose an approach to estimate the number of distinct n-grams, $1 \leq n \leq 6$, in any size *corpora*. It

Table III
The *best* (β, α) *combinations* and the vocabulary sizes found for English *corpora*

	1-grams	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
$\boldsymbol{\beta}$	7.7950	48.1500	21.8550	0.4200	-0.4400	0.6150
$\boldsymbol{\alpha}$	1.3466	1.1873	0.9800	0.8252	0.8000	0.8000
Voc Size	1.95×10^8	7.08×10^8	3.54×10^9	9.80×10^9	5.06×10^{10}	3.92×10^{11}

Table IV
The *best* (β, α) *combinations* and the vocabulary sizes found for French *corpora*

	1-grams	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
$\boldsymbol{\beta}$	16.4850	73.8550	71.3800	7.3500	-0.0700	2.4400
$\boldsymbol{\alpha}$	1.4496	1.2653	1.0444	0.8843	0.7835	0.8602
Voc Size	1.60×10^8	4.25×10^8	1.52×10^9	4.19×10^9	7.98×10^9	8.50×10^{10}

Table V
Relative errors of the estimated number of distinct n-grams for each English *corpus*. Values are in percentage (%)

***Corpus* Size**	**1-grams**	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
2,226,162	-2.9	-0.5	0.0	-0.1	0.5	-0.5
4,450,249	-0.3	-0.9	-0.7	-1.1	0.6	4.7
8,955,079	2.2	-0.9	-0.8	-0.8	0.6	-3.0
18,006,731	3.7	-0.5	-0.5	-0.5	1.0	-4.0
35,771,592	4.2	0.0	0.0	0.0	0.9	0.9
72,677,601	4.3	0.5	0.3	0.3	0.6	0.7
140,275,807	3.4	0.8	0.6	0.6	0.5	1.0
245,492,006	2.2	0.8	0.6	0.7	0.3	-0.3
490,846,877	-0.1	0.3	0.4	0.3	-0.1	-0.5
981,996,022	-2.9	-0.5	-0.3	-0.5	0.4	-0.5

Table VI
Relative errors of the estimated number of distinct n-grams for each French *corpus*. Values are in percentage (%)

***Corpus* Size**	**1-grams**	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
2,172,301	-1.8	-0.6	0.2	-0.5	-0.2	0.0
4,322,189	-0.7	-1.0	-0.3	-0.5	-1.0	2.7
8,710,051	0.5	-0.9	-0.4	-0.6	-0.6	0.4
17,442,724	1.1	-0.8	-0.2	-0.2	-0.3	1.1
34,744,534	1.6	-0.3	0.1	0.1	0.1	0.1
69,331,062	1.7	0.2	0.5	0.5	0.5	-0.4
139,025,258	1.6	0.8	0.8	0.8	0.8	-0.6
242,346,014	1.1	1.0	0.9	0.9	0.8	-0.7
484,314,987	0.6	1.5	0.8	0.8	0.7	-0.4
970,351,308	-1.7	0.9	-0.5	-0.4	-0.3	0.2

Table VII
Plateau values (vocabulary sizes) for distinct English n-grams and corresponding *corpus* size thresholds

	1-grams	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
***Plateau* Value**	1.95×10^8	7.08×10^8	3.54×10^9	9.80×10^9	5.06×10^{10}	3.92×10^{11}
***Corpus* Size Threshold**	9.22×10^{11}	1.05×10^{12}	1.29×10^{12}	1.43×10^{12}	6.18×10^{12}	2.39×10^{13}

Table VIII
Plateau values (vocabulary sizes) for distinct French n-grams and corresponding *corpus* size thresholds

	1-grams	**2-grams**	**3-grams**	**4-grams**	**5-grams**	**6-grams**
***Plateau* Value**	1.60×10^8	4.25×10^8	1.52×10^9	4.19×10^9	7.98×10^9	8.50×10^{10}
***Corpus* Size Threshold**	8.78×10^{11}	9.69×10^{11}	1.12×10^{12}	1.20×10^{12}	1.43×10^{12}	6.80×10^{12}

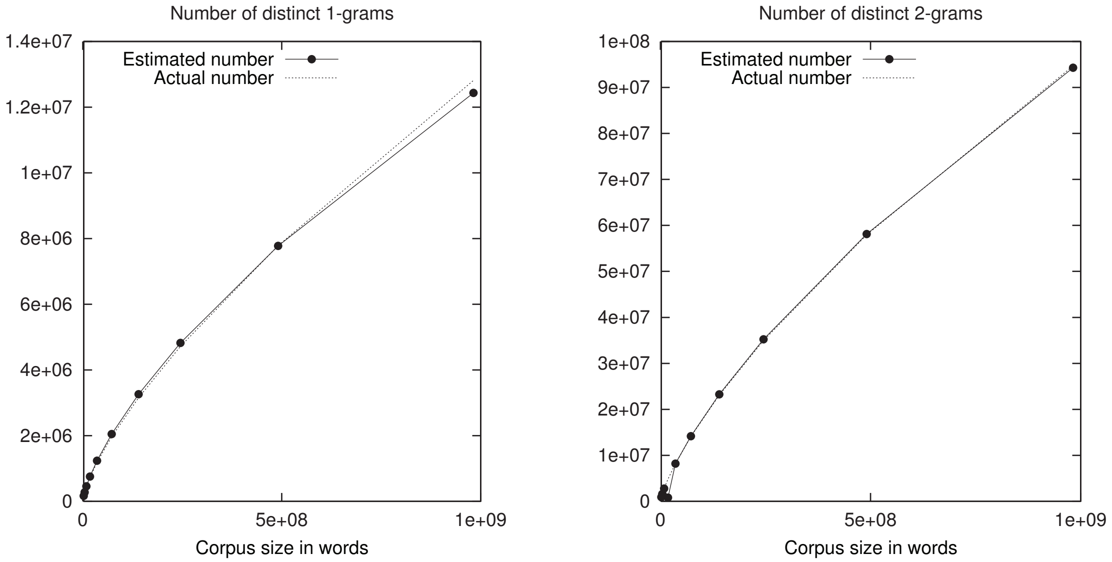

Figure 1. Actual *versus* estimated number of distinct 1-grams and 2-grams for English *corpora*

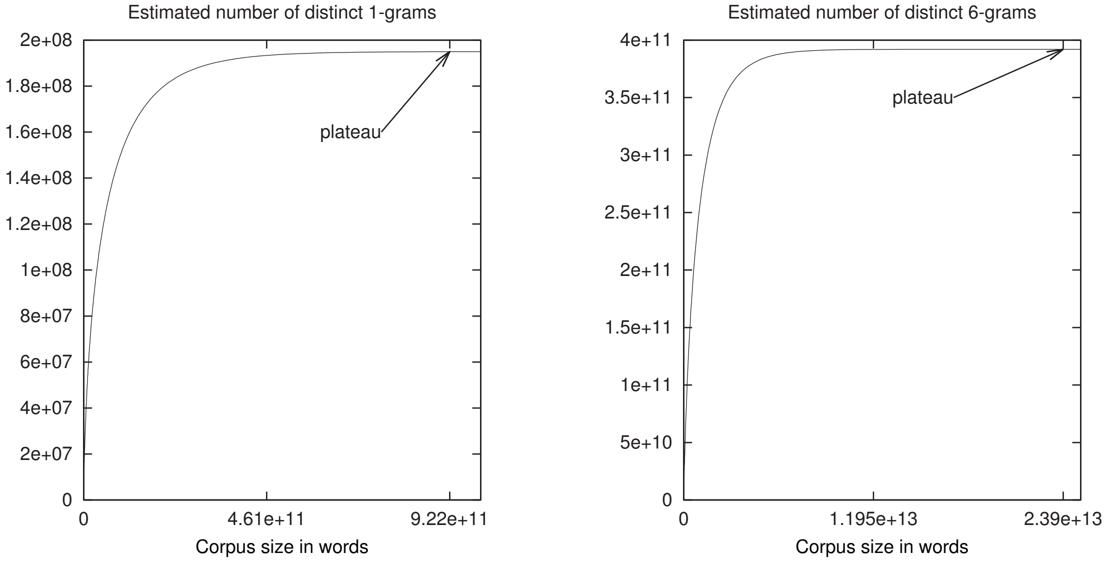

Figure 2. Estimated number of distinct 1-grams and 6-grams for Big Data English *corpora*

can be used for English or other languages, as long as a small and a large *corpus* can be used to tune the parameters for that language and for each n-gram size. The approach is based on Zipf-Mandelbrot Law and uses the Poisson distribution. Computationally heavy sums are replaced with an integral in order to provide high performance calculation, which can be useful for Big Data applications in the context of data mining, database systems or for cache size and hashing size calculation, where the memory space to accommodate cardinalities needs to be quickly estimated.

In the context of this development, we noticed that the probability of each n-gram tends to remain constant when *corpora* of the same language grow in size, which became evident for frequent n-grams, in the experiments we made. And there is no reason to think that, for Big Data *corpora*, the same will not happen for less frequent n-grams. This property led us to develop this approach.

Although vocabularies for each n-gram size are open, as new words may arise and others tend to stop being used, they are finite. This sets a *plateau* for the maximum number of distinct n-grams that any *corpus* can have, as our approach shows when estimates are calculated for Big Data *corpora*.

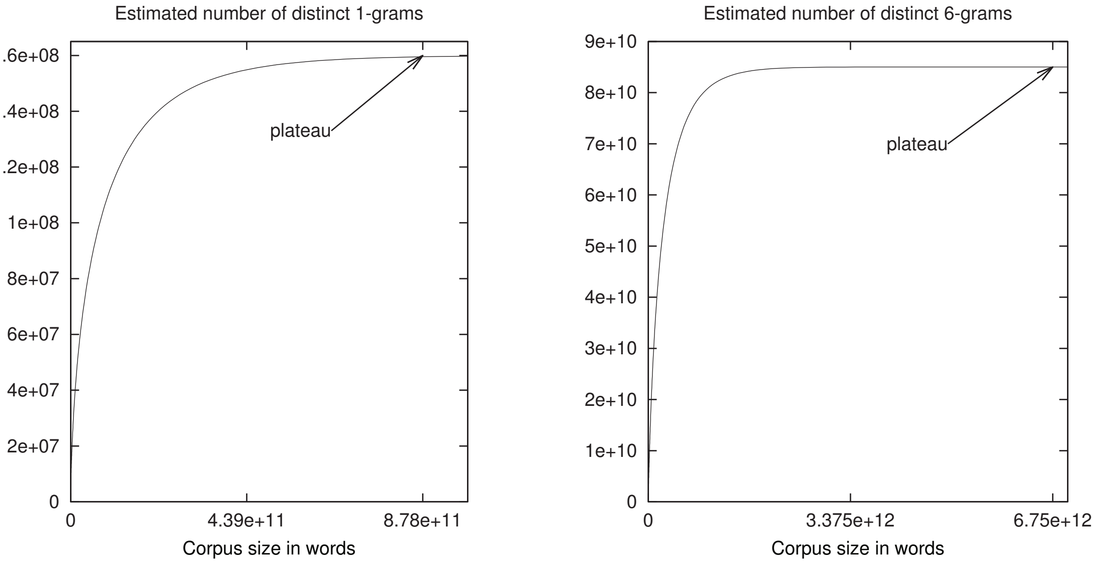

Figure 3. Estimated number of distinct 1-grams and 6-grams for Big Data French *corpora*

Tests showed promising results for the calculations of estimates, as the highest relative error was lower than 5%.

Acknowledgment

Thanks are due to FCT MCTES and NOVA LINCS UID/CEC/04516/2013.

References

[1] G. Zipf, *The Psychobiology of Language: An Introduction to Dynamic Philology*. Cambridge, Mass.: M.I.T. Press, 1935.

[2] B. Mandelbrot, "On the theory of word frequencies and on related markovian models of discourse," in *Structure of Language and its Mathematical Aspects*, vol. XII, 1953, pp. 190–210.

[3] J. B. Carroll, "On sampling from a lognormal model of word frequency distribution," *Computational analysis of present-day American English*, vol. 16, no. 3, pp. 406–424, 1967.

[4] G. K. Zipf, *Human Behavior and the Principle of Least-Effort*. Cambridge, MA: Addison-Wesley, 1949.

[5] D. Y. Manin, "Mandelbrot's model for Zipf's law: Can Mandelbrot's model explain Zipf's law for language?" *Journal of Quantitative Linguistics*, vol. 16, no. 3, pp. 274–285, 2009.

[6] S. T. Piantadosi, "Zipf's word frequency law in natural language: A critical review and future directions," *Psychonomic Bulletin & Review*, vol. 21, pp. 1112–1130, 2014.

[7] L. Egghe, "Untangling Herdan's law and Heaps' law: Mathematical and informetric arguments." *JASIST*, vol. 58, no. 5, pp. 702–709, 2007.

[8] D. C. van Leijenhorst and T. P. van der Weide, "A formal derivation of Heaps' law," *Information Sciences*, vol. 170, no. 2-4, pp. 263–272, 2005.

[9] A. Kornai, "Zipf's law outside the middle range," in *Proceedings of the Sixth Meeting on Mathematics of Language (MOL), University of Central Florida*, 1999, pp. 347–356.

[10] R. A. Baeza-Yates and G. Navarro, "Block addressing indices for approximate text retrieval," *Journal of the American Society of Information Science*, vol. 51, no. 1, pp. 69–82, 2000.

[11] D. Lemire and O. Kaser, "Recursive n-gram hashing is pairwise independent, at best," *Computer Speech & Language*, vol. 24, no. 4, pp. 698–710, 2010.

[12] ——, "One-pass, one-hash n-gram statistics estimation," *CoRR*, vol. abs/cs/0610010, 2006.

[13] J. A. Thom and J. Zobel, "A model for word clustering." *JASIS*, vol. 43, no. 9, pp. 616–627, 1992.

[14] F. A. Haight, *Handbook of the Poisson Distribution*. New York: John Wiley & Sons, 1967.

[15] M. Abramowitz, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. 10th printing*. New York: Stegun, Irene A., eds., 1972.

[16] Wikipedia. (2016, February) Wikimedia downloads, https://dumps.wikimedia.org/.

[17] J. F. D. Silva and G. P. Lopes, "A local maxima method and a fair dispersion normalization for extracting multiword units," in *In Proceedings of the 6th Meeting on the Mathematics of Language*, 1999, pp. 369–381.

[18] C. Gonçalves, J. F. da Silva, and J. C. e Cunha, "An n-gram cache for large-scale parallel extraction of multiword relevant expressions with LocalMaxs," in *To appear in 12th IEEE International Conference on eScience*, October 2016.