Entropy **2010**, *12*, 1743-1764; doi:10.3390/e12071743

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

Fitting Ranked Linguistic Data with Two-Parameter Functions

Wentian Li [1,⋆], **Pedro Miramontes** [2,4] **and Germinal Cocho** [3,4]

[1] Feinstein Institute for Medical Research, North Shore LIJ Health Systems, 350 Community Drive, Manhasset, NY 11030, USA

[2] Departamento de Matemáticas, Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Exterior, Ciudad Universitaria, México 04510 DF, Mexico; E-Mail: pmv@ciencias.unam.mx

[3] Departamento de Sistemas Complejos, Instituto de Física, Universidad Nacional Autónoma de México, Apartado Postal 20-364, México 01000 DF, Mexico; E-Mail: cocho@fisica.unam.mx

[4] Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Circuito Escolar, Ciudad Universitaria, México 04510 DF, Mexico

⋆ Author to whom correspondence should be addressed; E-Mail: wli@nslij-genetics.org; Tel.: +1 516 562 1076; Fax: +1 516 562 1153.

Received: 4 April 2010; in revised form: 18 May 2010 / Accepted: 1 July 2010 / Published: 7 July 2010

Abstract: It is well known that many ranked linguistic data can fit well with one-parameter models such as Zipf's law for ranked word frequencies. However, in cases where discrepancies from the one-parameter model occur (these will come at the two extremes of the rank), it is natural to use one more parameter in the fitting model. In this paper, we compare several two-parameter models, including Beta function, Yule function, Weibull function—all can be framed as a multiple regression in the logarithmic scale—in their fitting performance of several ranked linguistic data, such as letter frequencies, word-spacings, and word frequencies. We observed that Beta function fits the ranked letter frequency the best, Yule function fits the ranked word-spacing distribution the best, and Altmann, Beta, Yule functions all slightly outperform the Zipf's power-law function in word ranked-frequency distribution.

Keywords: Zipf's law; regression; model selection; Beta function; letter frequency distribution; word-spacing distribution; word frequency distribution; weighting

---

1. Introduction

Empirical linguistic laws tend to be functions with one key parameter: Zipf's law [1], which describes the relationship between the number of occurrence of a word (f) and the resulting ranking of the word from common to rare (r), states that $f \sim 1/r^a$, with the scaling exponent a as the free parameter. Herdan's or Heaps' law describes how the number of distinct words used in a text (V, or "vocabulary size", or number of types) increases with the total number of appearance of words (N, or, the number of tokens) as: $V \sim N^\theta$ [2,3]. Although in both Zipf's and Herdan's law, there is a second parameter: $f = C/r^a, V = CN^\theta$, the parameter C is either a y-intercept in linear regression and thus not considered to be functionally important, or subtracted from the total number of parameters by the normalization constraint (e.g., $\sum_i (f_i/N) = 1$).

Interestingly, one empirical linguistic law, the Menzerath-Altmann law [4,5], concerning the relationship between the length of two linguistic units, contains two free parameters. Suppose the higher linguistic unit is the sentence, whose length (y) is measured by the number of the lower linguistic unit, words, and the length of words (x) is measured by the even lower linguistic unit, e.g., phonemes or letters. Then, the Menzerath-Altmann law states that $y \sim x^b e^{-c/x}$ with two free parameters, b and c. Few people paid attention to the two-parameter nature of this function as it does not fit the data as good as the Zipf's law on its respective data, although Baayen listed several multiple-parameter theoretical models in his review on word frequency distribution [6].

We argue here about need to apply two-parameter functions to fit ranked linguistic data for two reasons. First, some ranked linguistic data, such as the letter usage frequencies, do not follow well a power-law trend. It is then natural to be flexible in data fitting by using two-parameter functions. Second, even for the known cases of "good fits", such as the Zipf's law on ranked word-frequency distribution *i.e.*, rank-frequency plot), the fitting tends to be not so good when the full ranking range is included. This imperfection is more often ignored than investigated. We would like to check whether one can model the data even better than Zipf's law by using two-parameter fitting functions.

The first 2-parameter function we consider is the Beta-function which attempts to fit the two ends of a rank-frequency distribution by power-laws with different exponents [7,8]. Suppose the variable f values are ranked: $f_{(1)} \geq f_{(2)} \geq f_{(3)} \cdots \geq f_{(n)}$, we define the normalized frequencies $p_{(r)} \equiv f_{(r)}/N$ such that $\sum_{r=1}^{n} f_{(r)} = 1$, ($n$ denotes the number of items to be ranked, r denotes the rank number, and $N = \sum_{r=1}^{n} f_{(r)}$ is the normalization factor). In a Beta function, the ranked distribution is modeled by

$$\text{Beta:} \qquad p_{(r)} = C\frac{(n+1-r)^b}{r^a} \tag{1}$$

where the parameter a characterizes the scaling for low-rank-number (*i.e.*, high frequency) items points, and b characterizes the scaling for the high-rank-number (*i.e.*, low frequency) items points. For the example of English words, n is the vocabulary size, and "the" is usually the $r = 1$ (most frequent) word.

If a logarithmic transformation is applied to both sides of Equation (1), the Beta function can be cast in a multiple regression model:

$$\text{Beta (linear regression form):} \qquad \log(p_{(r)}) = c_0 + c_1 \log(r) + c_2 \log(r') \tag{2}$$

where $r' = n + 1 - r$, $c_1 = -a$, $c_2 = b$, and $c_0 = \log(C)$.

The second 2-parameter function we are interested in is the Yule-function [9]:

$$\text{Yule:} \qquad p_{(r)} = C\frac{b^r}{r^a} \tag{3}$$

A previous application of Yule's function to linguistic data can be found in [10].

The Menzerath-Altmann function mentioned earlier:

$$\text{Menzerath-Altmann:} \qquad p_{(r)} = Cr^b e^{-a/r} \tag{4}$$

cannot be guaranteed to be a model for ranked distribution as the monotonically decreasing property is not always true even when the parameter b stays negative. Note that Menzerath-Altmann function is a special case of $f = Cr^b exp(-a/r - d \cdot r)$ which shares the same functional form as the generalized inverse Gauss-Poisson distribution [6].

Another two-parameter function, proposed by Mandelbrot [11], cannot be easily cast in a regression framework, and it is not used in this paper:

$$\text{Mandelbrot:} \qquad p_{(r)} = \frac{A}{(r+b)^a} \tag{5}$$

For random texts, one can calculate the value of b, e.g., $b = 26/25 = 1.04$ for 26-alphabet languages [12]. Clearly, all these 2-parameter functions one way or the other attempt to modify the power-law function:

$$\text{power-law, Zipf:} \qquad p_{(r)} = \frac{C}{r^a} \tag{6}$$

For comparison purposes, the exponential function is also applied:

$$\text{exponential:} \qquad p_{(r)} = Ce^{-ar} \tag{7}$$

Two more fitting functions, whose origin will be explained in the next section, are labeled as Gusein-Zade [13] and Weibull functions [14]:

$$\text{Gusein-Zade:} \qquad p_{(r)} = C\log\left(\frac{n+1}{r}\right) \tag{8}$$

$$\text{Weibull:} \qquad p_{(r)} = C\left(\log\left(\frac{n+1}{r}\right)\right)^a \tag{9}$$

In this paper, these functions will be applied to ranked letter frequency distributions, ranked word-spacing distributions, and ranked word frequency distributions. A complete list of functions, their corresponding multiple linear regression form, and the dataset each function will be applied, are included in Table 1. When the normalization factor N is fixed, parameter C is no longer freely adjustable, and the number of free parameters, $K-1$, is one less the number of fitting parameters, K. The value of $K-1$ is also listed in Table 1.

The general framework we adopt in comparing different fitting functions is Akaike information criterion (AIC) [15] in regression models. In regression, model parameters in model $y = F(x)$ are estimated to minimize the sum of squared errors (SSE):

$$SSE = \sum_{i=1}^{n}(y_i - F(x_i))^2 \tag{10}$$

It can be shown that when the variance of the error is unknown, which is to be estimated from the data, least square leads to maximum likelihood $\hat{L}$ (the underlying statistical model is that the fitting noise or deviance is normally distributed) (p. 185 of [16]):

$$\hat{L} = C - \frac{n}{2}\log\frac{SSE}{n} - \frac{n}{2} \tag{11}$$

The model selection based on AIC minimizes the following term:

$$AIC = -2\hat{L} + 2K \tag{12}$$

where K is the number of fitting parameters in the model. Combining the above two equations, it is clear that AIC of a regression model can be expressed by SSE (any constant term will be canceled when two models are compared, so it is removed from the definition):

$$AIC = n\log\frac{SSE}{n} + 2K \tag{13}$$

For two models applied to the same dataset, the difference between their AIC's is:

$$AIC_2 - AIC_1 = n\log\frac{SSE_2}{SSE_1} + 2(K_2 - K_1) \tag{14}$$

If model-2 has one more parameter than model-1, model-2 will be selected if $SSE_2/SSE_1 < e^{-2/n}$. Note that almost all fitting functions listed in Table 1 are variations of the power-law function, so the regression model is better applied to the log-frequency scale: $y = \log(p)$.

Table 1. List of functions discussed in this paper. The number of free parameters in a function is $K-1$. The functional form in the third and the 4th column is for $p_{(r)} = f_{(r)}/N$, or $\log(p_{(r)})$, respectively. We define: $x_1 = \log(r)$, $x_2 = \log(n+1-r)$, $x_3 = 1/r$, and $x_4 = \log(\log((n+1)/r))$. The last 3 columns indicate which dataset a function is applied to (marked by x): dataset 1 is the ranked letter frequency distribution, dataset 2 is the ranked inter-word spacing distribution, and dataset 3 is the ranked word frequency distribution.

model	$K-1$	formula ($p_{(r)} = f_{(r)}/N$)	linear regression ($y = \log p_{(r)}$)	data1 (letter freq.)	data2 (word spacing)	data3 (word freq.)
Gusein-Zade	0	$C\log((n+1)/r)$	$c_0 + x_4$	x	x	-
Weibull	1	$C(\log((n+1)/r))^a$	$c_0 + c_4 x_4$	x	x	-
power-law	1	C/r^a	$c_0 + c_1 x_1$	x	x	x
exponential	1	Ce^{-ar}	$c_0 + c_r r$	x	x	-
Beta	2	$C(n+1-r)^b/r^a$	$c_0 + c_1 x_1 + c_2 x_2$	x	x	x
Yule	2	Cb^r/r^a	$c_0 + c_r r + c_1 x_1$	x	x	x
Altmann	2	$Cr^b e^{-a/r}$	$c_0 + c_1 x_1 + c_3 x_3$	x	-	x
Mandelbrot	2	$C/(r+b)^a$	-	-	-	-

Before we fit and compare various functions, we would like to explain the origin of Gusein-Zade and Weibull function by showing the equivalence between an empirical ranked distribution and the empirical cumulative distribution in the next section.

2. Equivalence between an Empirical Rank-Value Distribution (eRD) and an Empirical Cumulative Distribution (eCD)

Our usage of rank-frequency distribution may raise a question of "why not use a more standard statistical distribution?" It is tempting to relate a sample-based (empirical) ranked distribution to the "order statistics" [17]. However, they are not equivalent. Suppose a particular realization of the values of n variables being ranked (ordered) such that: $x_{(1)} > x_{(2)} > x_{(3)} \cdots > x_{(n)}$, and there are many (e.g., m) realizations; the distribution of m top-ranking (rank-1, order-1, maximum) variable values ($\{x^i_{(1)}\}, i = 1, 2, \cdots m$) characterize an order statistic an empirical rank-value distribution (x-axis: $(1, 2, \cdots n)$, y-axis: $(x_{(1)}, x_{(2)}, \cdots x_{(n)})$) only characterizes the distribution of one particular realization.

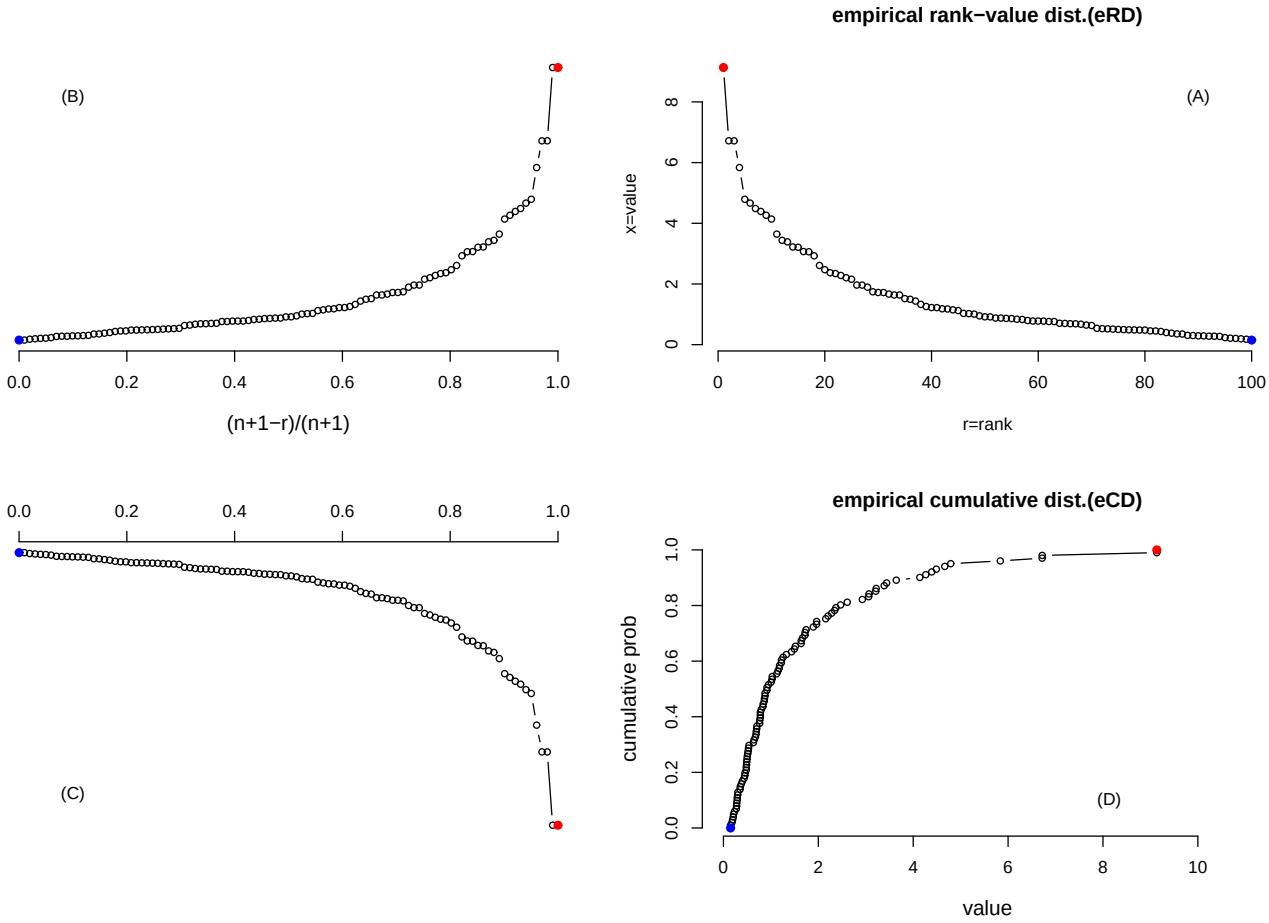
However, empirical rank-value distribution (eRD) is equivalent to the empirical cumulative distribution (eCD) of the n values of variable x from a data set. Figure 1 shows a step-to-step transformation (starting from the upper-right, counter-clockwise) from an eRD to an eCD, using $n = 100$ values randomly sampled from the log-normal distribution. One empirical rank-value distribution in Figure 1(A) is transformed to Figure 1(B) by a mirror image with respect to the y-axis, plus a transformation of the x-axis: the new x-axis is reversed from the old x-axis and is normalized to (0,1). The purpose of this operation is to start the empirical cumulation from the lowest ranking point.

Figure 1(C) is a mirror image of Figure 1(B) with respect to x-axis, and Figure 1(D) is a counter-clockwise rotation of Figure 1(C) of $90°$. Figure 1(D) is then the empirical cumulative distribution of this data set, which shows the proportion of values that are smaller than a threshold marked by the x-axis. Actually, there is a simpler operation from Figure 1(A) to Figure 1(D): a clockwise rotation of $90°$ plus a reverse and a normalization of the old x-axis to the new y-axis in the range of (0,1).

The relationship between eRD and eCD has been discussed in the literature a number of times [18–23]. The reason we emphasize this relationship is to reinforce the understanding that a functional fitting of the rank-value distribution will correspond to a mathematical description of the empirical cumulative distribution, and vice versa. Since the derivative of the empirical cumulative distribution is the empirical probability density (ePD), we then have an estimation of ePD as well, even though it may not be in a close form. The relationship between an ePD and the true underlying distribution is always tricky [6,24].

The frequency spectrum (FS) [6] is defined as the number of different events occurring with a given frequency. The cumulative sum of FS, called empirical structural type distribution (eSTD), is closely related to the empirical cumulative distribution. Generally speaking, eCD can be converted to eSTD by mirroring with respect to the $y = 1$ line, then rescale the new y-axis from 1 to n (see Figure 1(D)). The situation is slightly more complicated when there are horizontal bars in eRD (equivalent to vertical lines in eCD) (see section 5). In that case, a horizontal bar is coverted to a point by its right end, before the mirror/rotation is carried out. We note that Alfred James Lotka used FS in his discovery of the power-law pattern in data [22,25].

Figure 1. (A) $n = 100$ log-normal distributed values being ranked (x-axis is the rank r, $r = 1$ for the largest value, $r = n$ for the smallest value, y-axis is the value itself); (B) Mirror image of (A) with respect to the y-axis. Note that the new x-axis is both reversed and normalized (so that lowest ranking value at 0 and highest ranking value at 1); (C) Mirror image of (B) with respect of x-axis; (D) Rotation of (C) of $-90°$. The highest ranking value is marked by the red color and the lowest ranking value by the blue color.



The Gusein-Zade and Weibull function introduced in section 1 can be understood by a reverse transformation from eCD to eRD. Suppose an eCD converges to 1 exponentially, which also describes the gap length distribution of a Poisson process with the mean gap length of f_m:

$$CD = 1 - e^{-\frac{f}{f_m}} \tag{15}$$

Similar to the survival curve in biostatistics [26], eCD is a step function with n vertical jumps. The lowest-ranking point ($r = n$) corresponds to the first step, with cumulative probability of $1/(n+1)$, and the highest-ranking point ($r = 1$) corresponds to the last step with the cumulative probability of $n/(n+1)$. This can be used to rewrite Equation (15) as:

$$\frac{n+1-r}{n+1} = 1 - e^{-\frac{f_{(r)}}{f_m}} \tag{16}$$

or,

$$f_{(r)} = f_m \log\left(\frac{n+1}{r}\right) \tag{17}$$

which is exactly Equation(8). In data fitting, we treat any coefficient in the function as a parameter, thus f_m/N becomes C. This rank-frequency distribution is exactly the same as the one proposed by Sabir Gusein-Zade [13,27,28].

Suppose the CD approaches 1 in a stretched exponential relaxation [29,30]:

$$CD = 1 - e^{-\left(\frac{f}{f_m}\right)^{\alpha}} \tag{18}$$

the converted RD is the Weibull function defined in Equation(9) with $C = f_m/N$ and $a = 1/\alpha$.

In general, any cumulative distribution of f

$$\frac{n+1-r}{n+1} = CD(f) \tag{19}$$

can be converted to RD by solving f as a function of r.

3. Ranked Letter Distribution

The frequency of letters in a language is of fundamental importance to communication and encryption. For example, the Morse code is designed so that the most common letters have the shortest codes. Shannon used the frequencies of letters as well as other units to study the redundancy and predictability of languages [31]. It is well known that the letter "e" is the most frequent letter in English [32,33]. On the other hand, the most common letter in other languages can be different, e.g., "a" for Turkish, and "a", "e", "i" share the top ranks in Italian, *etc*. (http://en.wikipedia.org/wiki/Letter_frequency)

We plot the ranked letter frequencies in the text of *Moby Dick* in Figure 2 both in linear-linear and log-log scales (see Appendix A for the source information). The number of bits in Morse code for these letters is also listed in Figure 2(A), which does show the tendency of low number of bits for more commonly used letters. Seven fitting functions (one zero-parameter, three one-parameter, and three two-parameter functions) are applied (Table 1). All can be framed in multiple regression (see Appendix B for programming information).

The fitting of Gusein-Zade equation in a regression for the log-frequency data needs more explanation. From Table 1, we see that there is no regression coefficient for the $x_4 = \log\log(27/r)$ term, and the only parameter to be estimated is the constant term. To show this fact, the regression model is re-written as:

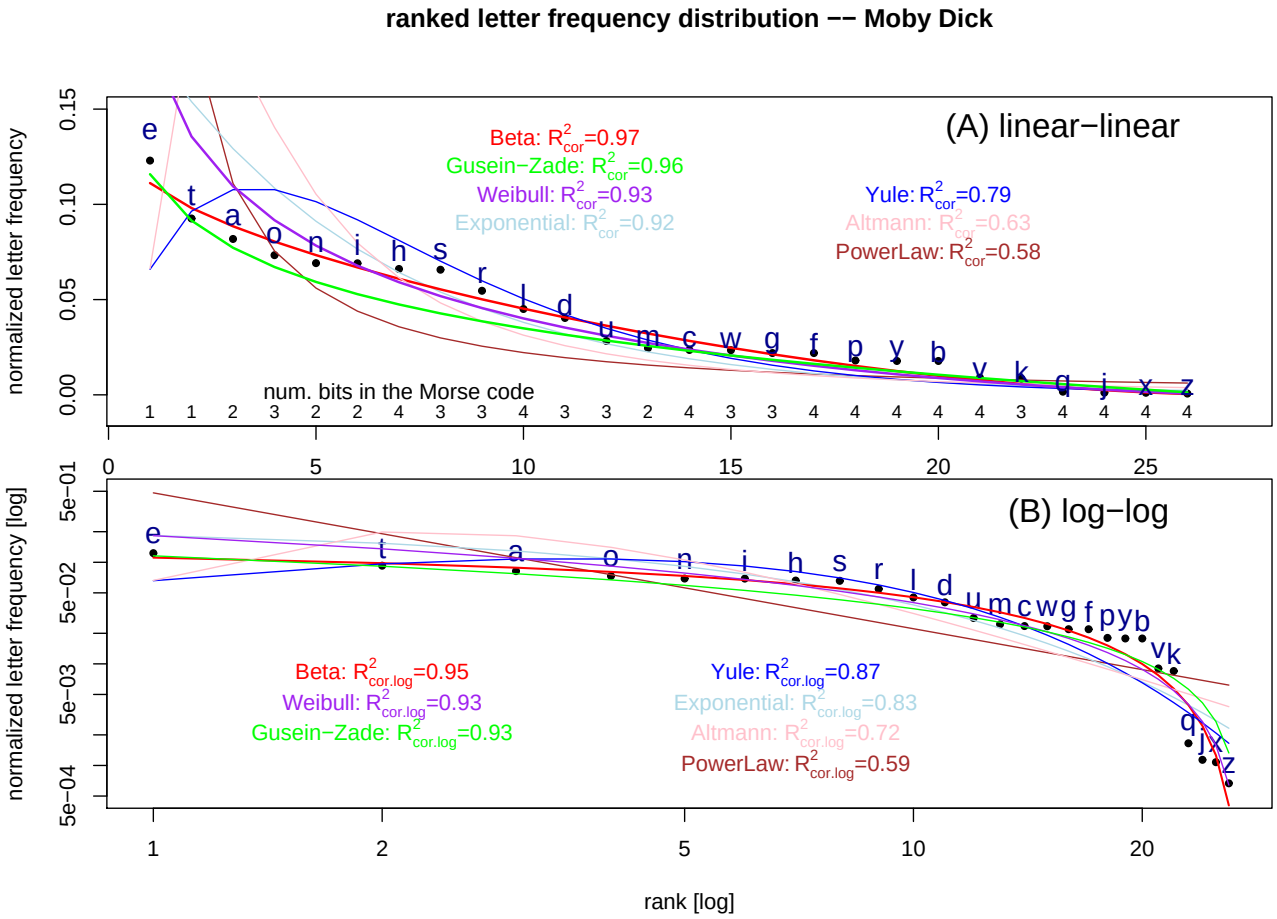
$$\log(p_{(r)}) - x_4 = \log\frac{p_{(r)}}{\log((n+1)/r)} = c_0 \tag{20}$$

From Figure 2, it is clear that power-law, Altmann, and Yule functions do not fit the data as well as other functions. When the one-parameter functions are compared, exponential function clearly out-performs the power-law function.

For regression on the log-frequency data, SSE and AIC can be determined, and conclusion on which model is the best model can be reached. Table 2 shows that Beta function is the best fitting function, followed by Weibull and Gusein-Zade function. The performance of Gusein-Zade function is impressive, considering that it has 2 fewer parameters than Beta and Weibull functions.

Another commonly used measure of fitting performance is R^2, which does ignore the issue of number of parameters. R^2 can either be defined as the proportion of variance "explained" by the regression or defined as the squared correlation between the observed and the fitted values (see Appendix C for more details).

Figure 2. Ranked letter frequency distribution for the text of *Moby Dick* (normalized so that the sum of all letter frequencies is equal to 1: $p_{(r)} = f_{(r)}/\sum_i f_{(r)}$). (A) in the linear-linear scale, and (B) in log-log scale. Seven fitting functions (see Table 1) are overlapped on the data: power-law function (brown, $a = 1.34$), exponential (lightblue, $a = 0.175$), Beta function (red, $a = 0.0886, b = 1.64$), Yule function (blue, $a = -0.938, b = 0.764$), Gusein-Zade (green), Weibull (purple, $a = 1.27$), and Altmann (pink, $a = 5.78, b = -2.58$). The number of bits in Morse code for the corresponding letter is written at the bottom of (A). The R^2 values listed are R^2_{cor} or $R^2_{cor,log}$ in Equation (26).



The R^2 values in Table 2 show again that Beta function is the best fitting function, followed by Weibull and Gusein-Zade functions. The R^2_{log} and $R^2_{cor,log}$ for Gusein-Zade function is not equal because the variance decomposition does not hold true (Appendix C) when Equation (20) is converted back to the form of $\log(p_{(r)}) = c_0 + x_4$.

When the regression is carried out in log-frequency scale, whereas the SSE and R^2 is calculated in the scale, we do not expect the variance decomposition holds true, and do not expect R^2 to be equal to R^2_{cor}. The last two columns in Table 2 shows SSE and R^2_{cor} for these models when y axis in linear scale. There are a few minor changes in the relative fitting performance among different functions. However, we caution in reading too much in these numbers as the parameter values are fitted to ensure the best performance in the log-frequency scale, not in the frequency scale.

Table 2. Comparison of regression models in fitting ranked letter frequency distribution obtained from the text of *Mody Dick*. The regression is applied to the log-frequency scale (Table 1). SSE: sum of squared error (Equation (10)); ΔAIC: Akaike information criterion relative to the best model (whose AIC is set at 0) (Equation (14)); R^2_{log}: variance ratio Equation (25); $R^2_{corr,log}$: an alternative definition of R^2 based on correlation (Equation (26)). The SSE and R^2_{cor}, based on the regression in the logarithmic scale, whereas applied to the linear y scale, are shown in the last two columns.

model	$K-1$	log-scale				linear-scale	
		SSE	Δ AIC	$R^2_{cor,log}$	R^2_{log}	SSE	R^2_{cor}
Gusein-Zade	0	5.77	16.9	0.933	0.892	.00196	0.962
power-law	1	22.2	53.9	0.586	0.586	0.145	0.580
exponential	1	8.91	30.2	0.834	0.834	0.0124	0.918
Weibull	1	3.59	6.52	0.933	0.933	.00726	0.928
Beta	2	2.58	0	0.952	0.952	.000716	0.973
Yule	2	6.66	24.6	0.876	0.876	.00752	0.795
Altmann	2	14.83	47.4	0.723	0.723	0.0323	0.628

The fitted Beta function in Figure 2(B) is:

$$\log(p_{(r)}) = -7.5254 + 1.6355\log(27-r) - 0.08856\log(r) \tag{21}$$

i.e., the two parameters in Equation (1) are $a \approx 0.09$ and $b \approx 1.6$. In [34], the relative magnitude of a and b is used to measure the consistency to power-law behavior ($a >> b$ provides a confirming answer, and $a << b$ does not). Equation (21) indicates that the ranked letter frequency distribution in English is very different from a power-law function.

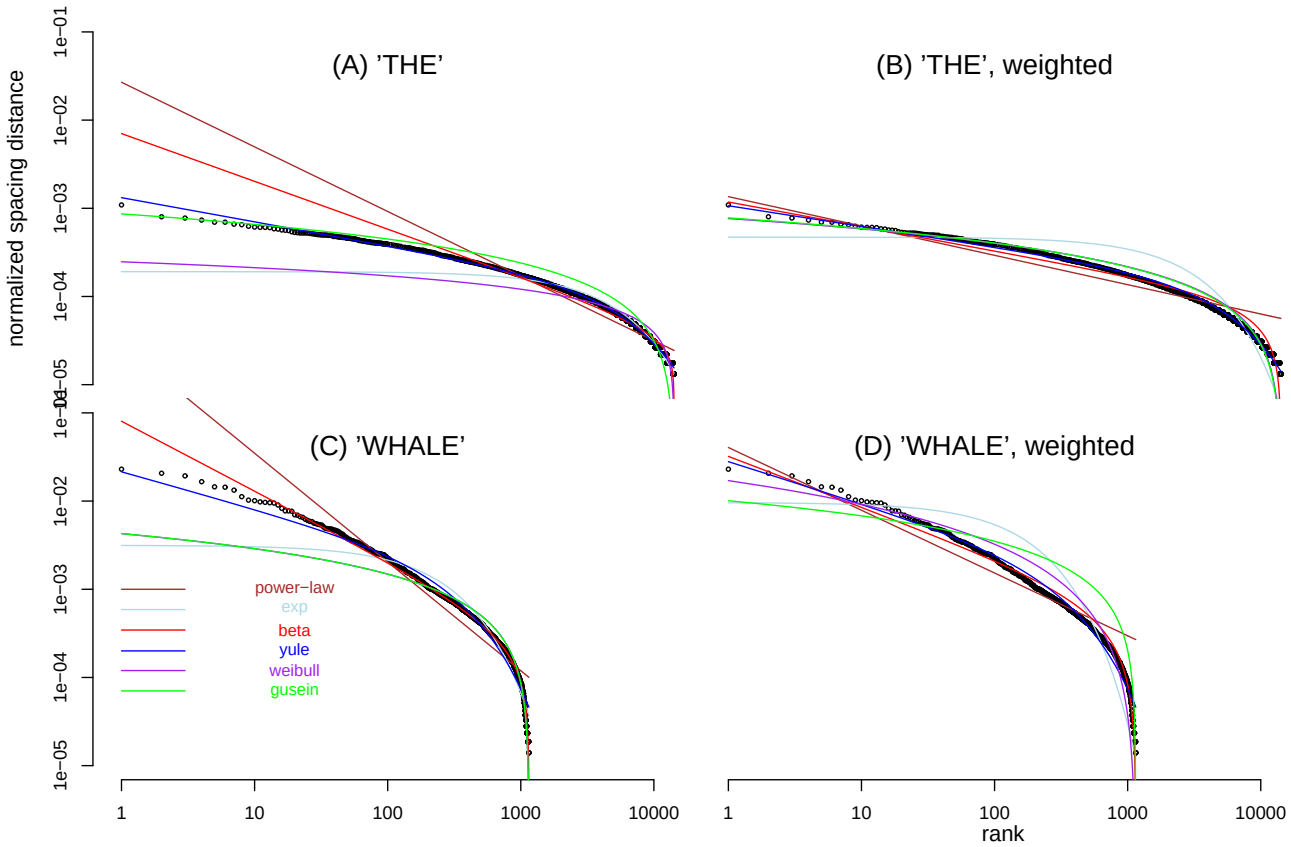
4. Ranked Inter-word Spacing Distribution

Motivated by level statistics in quantum disorder systems, it has been proposed that distance between successive occurrences of the same word might be related to whether or not that word plays an important role in the text (the so-called "keyword") [35,36]. Similar studies of gap distributions are also common in bioinformatics, such as the in-frame start-to-stop codon distances (which defines "open reading frames") [37,38] or in-frame stop-stop codon distances [39]. In the example provided in [36], for the text of *Don Quijote*, the function word "but" tends to be randomly scattered along the text, whereas the content word "Quijote" is more clustered. Here we will use the text of *Moby Dick* to obtain the ranked nearest-neighbor distance distributions of selected words, and check which functions fit the data.

Figure 3 shows two ranked nearest-neighbor spacing distributions (after normalization) between a given word, one for a function word and another for a content word. The representative of function words is "THE" which is the most common word used in *Moby Dick*. The representative of content words is "WHALE" which is the 23rd ranked word but the most common noun. The main conclusion in [35,36] is confirmed that function words such as THE tend to be evenly scattered whereas

content words such as WHALE tends to be clustered, leaving huge gaps between clusters (and the low-rank-number spacings are much larger).

Figure 3. Ranked nearest-neighbor spacing (normalized) between the word THE (in the unit of word) (A,B), and that between the word WHALE (C,D), in the text of *Moby Dick*. Both $x-$ and $y-$axis are in log scales. Six functions are used to fit the data: power-law (brown, $a = 0.333$ (THE, weighted), $a = 0.733$ (THE), $a = 0.712$ (WHALE, weighted), $a = 1.23$ (WHALE)), exponential (lightblue, $a = 0.000314$ (THE, weighted), $a = 0.000187$ (THE), $a = 0.0579$ (WHALE, weighted), $a = 0.00385$ (WHALE)), Beta (red, $a = 0.278, b = 0.665$ (THE, weighted), $a = 0.542, b = 0.295$ (THE), $a = 0.576, b = 0.980$ (WHALE, weighted), $a = 0.791, b = 0.673$ (WHALE)), Yule (blue, $a = 0.234, b = 0.9999$ (THE, weighted), $a = 0.274, b = 0.9999$ (THE), $a = 0.475, b = 0.997$ (WHALE, weighted), $a = 0.422, b = 0.997$ (WHALE)), Weibull (purple, $a = 0.982$ (THE, weighted), $a = 0.559$ (THE), $a = 1.57$ (WHALE, weighted), $a = 0.999$ (WHALE)), and Gusein-Zade functions (green), using either weighted least square regression (B,D) or regular least square regression (A,C). The weighting used is $w = 1/r$ (r is the rank).



When a power-law function is applied to fit the inter-word spacings, the fitting performance is not good. In particular, it fails badly to fit the low-rank-number points (see the brown line in Figure 3(A,C)). Since the density of points is denser at the tail of the ranking distribution, the least-square-fit regression result using all data points equally is not consistent with the visual impression in the log-log scale overlapping between the fitting line and the data.

There could be at least two solutions to this problem. The first solution is to use a subset of the data points so that these are evenly distributed in the log-transformed x-axis. The second solution is to assign a weight to each point in the least-square fitting, such that the weight is smaller when the density of data points is higher in the log-transformed x-axis. We adopt the second solution and choose a weight inversely proportional to the rank: $w = 1/r$. This particular weight is chosen by some trials-and-errors process, and no theoretical justification will be provided in this paper. Weighting is allowed in the R subroutine for linear regression (Appendix B).

When the weighting option is chosen in power-law function, the regression line does indeed seem to pass through the low-rank-number points (brown line in Figure 3(B,D)), although it still fails to fit the tail (high-rank number points) portion of the ranking distribution. Other functions, whose parameter values are determined by the weighted regression in log-spacing scale, are drawn in Figure 3(B,D) as well: exponential, Beta, Yule, Weibull, and Gusein-Zade function

Table 3 summarizes the fitting results of these functions by SSE and AIC. Two sets of result are presented: one with the weight $w = 1/r$ and the other without. Interestingly, Yule function is the best fitting function in most situations, for both the function word "the" and content word "whale". The Beta function is the close second best function. The AIC difference between models for weighted regression is calculated by the formula in Appendix E.

Table 3. Comparison of regression models in fitting ranked inter-word spacing distributions obtained from the text of *Moby Dick* for words THE and WHALE. The regression is applied to the log-distance scale (4th column of Table 1). Two sets of result are presented: one for the weighted regression (weight $w_r = 1/r$), another without weights. SSE: sum of squared error (Equation (10)); ΔAIC: Akaike information criterion relative to the best model (whose AIC is set at 0) (Equation (14)). One R^2 ($R^2_{cor,log,weight}$ in Equation (31)) is listed in the last column.

word	model	$K-1$	weighted regression		unweighted regression		$R^2_{cor,log,weight}$
			SSE	Δ AIC	SSE	Δ AIC	
	Gusein	0	0.701	25.9	5860.9	73588.2	0.933
T	power-law	1	0.947	31.0	1014.0	48720.0	0.909
H	exponential	1	2.49	40.8	295.1	31222.6	0.761
E	Weibull	1	0.698	27.9	1332.5	52591.9	0.933
	Beta	2	0.265	20.1	297.4	31334.1	0.975
	Yule	2	0.0366	0	32.6	0	0.996
	Gusein	0	3.83	18.7	155.2	2537.2	0.931
W	power-law	1	1.62	14.2	306.0	3320.2	0.919
H	exponential	1	3.58	20.2	122.4	2265.8	0.821
A	Weibull	1	1.38	12.9	155.2	2539.2	0.931
L	Beta	2	0.360	4.69	17.1	0	0.982
E	Yule	2	0.195	0	75.0	1703.2	0.990

Among several choices of R^2s (see Appendix C, D), we list $R^2_{cor,log,weight}$ in Table 3. The relative fitting performance judged by R^2 is consistent with that by SSE and AIC. It is worth noting that the highest R^2 values, which is achieved by the Yule function, $R^2 = 0.996$ for word THE, 0.990 for word WHALE, are quite high. It indicates that Yule function fits the data very well.

It is proposed in [35] that if a word is randomly distributed along the word sequence, it is a Poisson process and the nearest-neighbor spacing will follow an exponential distribution. That equation is exactly the Gusein-Zade function in Equation (8). Both Figure 3 and Table 3 shows, however, that Gusein-Zade function is not the best fitting function when compared to others.

5. Revisiting Zipf's Law

Ranked word frequency distribution in natural [1] and artificial languages [40] follows a reasonably good power-law, especially when only the low-rank-number words are fitted. Ranking words by their frequency of usage has practical applications, for example, it may establish the priorities in learning a language [41]. However, the fitting by a power-law function for the whole range of ranks is by no mean perfect. In [42], it was suggested that two power-law fitting with two different scaling exponents are needed to fit the ranked word frequencies. In a more striking illustration [43], 42 million words (tokens) from Wall Street Journal were used to draw the ranked word frequencies, and it deviate from the Zipf's law at $r \approx 5000$.

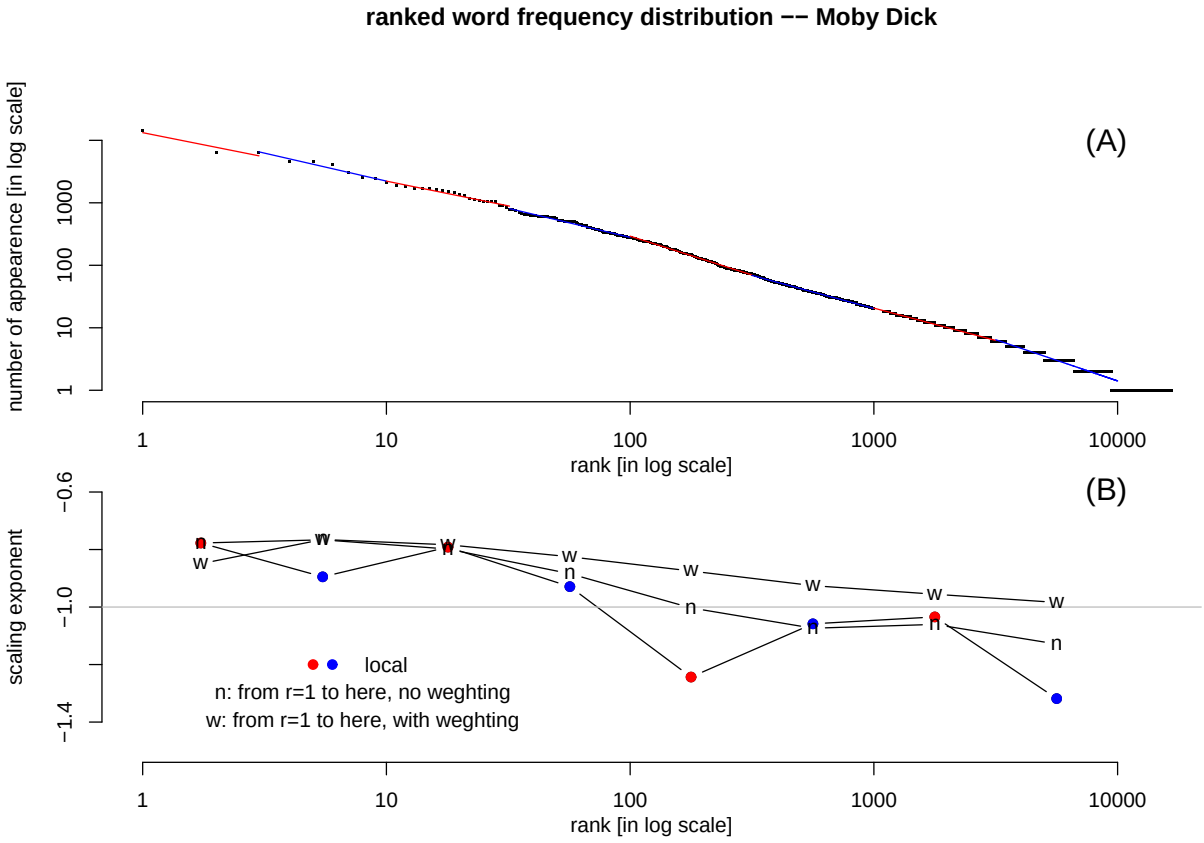
Figure 4(A) shows the ranked word frequency distribution for the text *Moby Dick*. In order to examine more closely how good Zipf's law fits the data, we carry about three regression analyses: one is a local regression using only points whose rank is within certain range; the second is a regression for all points whose ranking number is lower ($r < r_0$) than certain value. That regression is repeated with the weight of $w = 1/r$.

The scaling exponents from these three types of regressions are plotted in Figure 4(B). It is clear that the slope is becoming steeper towards the tail (high-rank-number words). The regression with weight manages to slow down the increase of the slope, as the points near the tail are weighted down. Still, the slope is not a constant. The slope of -1 as expected by the Zipf's law is marked in Figure 4(B) as a reference. Figure 4 confirms the previous observations that Zipf's law is not a perfect fit of the ranked word frequency distribution in the full rank range [42,43].

This leaves room for improvement by a two-parameter fitting function. Figure 5 shows the results of three 2-parameter fitting functions (all with weight $w = 1/r$) plus that of power-law function, one for words frequency in *Moby Dick* and another for *Don Quijote* (see Appendix A for source information). Exponential function, Gusein and Weibull function all fit the data badly, so these are not included in the plot.

Table 4 shows SSE and ΔAIC (relative to the best model) of these four functions, in both weighted and unweighted regression. For weighted regression, Altmann function is the best model, although other three functions are not far behind. For unweighted regression, Beta function is the best model. The reason for this difference might be due to the ability of Altmann function to make nonlinear (in log-log plot) turn at the head area of the curve (low-rank-number), and that of the Beta function at the tail area (high-rank-number). Table 4 also shows that ranking of functions by $R^2_{cor,log,weight}$ values is consistent with that by SSE.

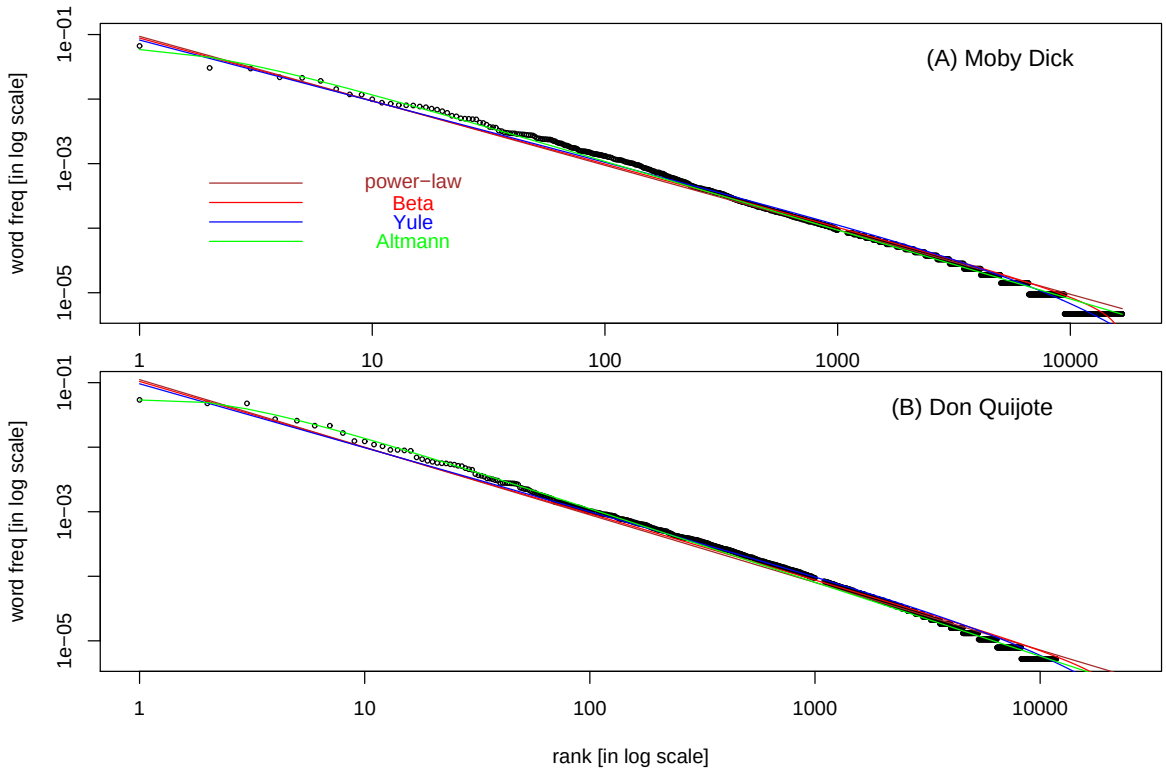
Figure 4. (A) Ranked word frequency distribution of text *Moby Dick*, in log-log scale. The number of word type is 16713, and the number token is 212791. Points within certain rank range are fitted with power-law locally, alternately colored by red and blue. (B) The slope of the local regression line (red/blue): the slope of regression for all points higher than certain rank ($r < r_0$): without weight ("n") and with weight ($w = 1/r$) ("w").



We may conclude from Table 4 that it is worthwhile to use two-parameter models, as all three (Beta, Yule, and Altmann function) outperform the Zipf's law, even if not by much. One important point made in [6] is that when dealing with the word frequency distribution, due to the presence of "large number of rare words" [6,24], the length of the text may affect the result. Consequently, it is preferable to check the obtained word statistics as a function of the sample size.

In order to check whether the conclusion that two-parameter models outperform Zipf's law is still true at a different sample size, we split the *Don Quijote* into two books: *El ingenioso hidalgo don Quijote de la Mancha*, and *Segunda parte del ingenioso caballero don Quijote de la Mancha*. The SSE and relative AIC result of these two books are shown in Table 4. The main conclusion that Altmann function is the best model in weighted regression, and Beta function is the best in unweighted regression, remain the same.

Figure 5. Ranked word frequency (normalized) in the English text of *Moby Dick* (MD) (A), and the Spanish text of *Don Quijote* (DQ) (B). The following weighted regressions are applied and the resulting fitting lines are shown (Table 1): power-law function or Zipf's law (brown, $a = 0.999$ (MD, weighted), $a = 1.05$ (DQ, weighted)), Beta function (red, $a = 0.975, b = 0.293$ (MD, weighted), $a = 1.02, b = 0.312$ (DQ, weighted)), Yule function (blue, $a = 0.946, b = 0.9999$ (MD, weighted), $a = 0.99, b = 0.9999$ (DQ, weighted)), and Menzerath-Altmann function (green, $a = 0.932, b = -1.07$ (MD, weighted), $a = 1.41$, $b = -1.15$ (DQ, weighted)).



6. Discussion

It is natural to add more parameters to models when they fail to fit satisfactorily certain sets of data (e.g., [44]). Nevertheless, if the mathematicians' task were only to fit the data, then with as many parameters as the points in the dataset it would be possible in principle to fit anything but the resulting model would be useless because it would be good only for a particular set of data. The goal should then be to find a balance between the extremes of too few parameters and bad fittings and too many parameters and overfitting.

For many linguistic data, one-parameter functions perform reasonably well, but when analyzing large datasets there are some discrepancies at the extremes of the abscissae. It is easy to understand why we would like to try some more parameters starting with two parameter models. These models have not been extensively used partly because of a lack of critical comparison among them, and partly because their performance may depend on particular dataset used.

Table 4. Comparison of regression models in fitting ranked word frequency distributions obtained from the English text of *Moby Dick*, and the Spanish text of *Don Quijote*. The two books of *Don Quijote* are also separated for additional analysis with the goal of checking sample-size effect. The regression is applied to the log-distance scale (4th column of Table 1). Two sets of result are presented: one for the weighted regression (weight $w_r = 1/r$), another without weights. SSE: sum of squared error (Equation (10)); ΔAIC: Akaike information criterion relative to the best model (whose AIC is set at 0) (Equation (14)). The last column is $R^2_{cor,log,weight}$ in Equation (31).

data	model	$K-1$	weighted regression		unweighted regression		$R^2_{cor,log,weight}$
			SSE	ΔAIC	SSE	ΔAIC	
Moby Dick	power-law	1	0.656	7.54	452.7	551.7	0.993
	Beta	2	0.523	7.21	438.0	0	0.994
	Yule	2	0.369	3.69	442.8	184.0	0.996
	Altmann	2	0.260	0	441.2	120.3	0.997
Don Quijote	power-law	1	1.10	17.9	783.8	6018.2	0.990
	Beta	2	0.950	18.3	603.4	0	0.991
	Yule	2	0.721	15.3	748.7	4959.5	0.993
	Altmann	2	0.170	0	771.5	5654.3	0.998
Don Quijote book-1	power-law	1	0.831	16.3	560.9	6045.8	0.991
	Beta	2	0.751	17.3	377.2	0	0.992
	Yule	2	0.592	14.8	483.0	3767.8	0.994
	Altmann	2	0.138	0	556.9	5937.7	0.999
Don Quijote book-2	power-law	1	0.964	19.0	591.2	6861.8	0.990
	Beta	2	0.899	20.3	391.0	0	0.991
	Yule	2	0.758	18.5	499.1	4053.0	0.992
	Altmann	2	0.125	0	588.7	6791.9	0.999

For ranked letter frequency data, because the range of abscissae is short, the difference between various fitting functions can be small. Only when the independent variable range is expanded, could we see a divergence among different functions.

In this paper, we only limit ourselves to regression models on the log-frequency-log-rank scale. And it is interesting to find Beta function outperform the Gusein-Zade function traditionally applied in the letter frequency distribution [27]. Further analyses are needed, both by using other datasets and by using regression in the frequency scale, to have more confidence of the conclusion reached here.

For inter-word spacing distribution, if a word appears in the text randomly, its probability distribution is a geometric distribution [35]. Geometric distribution is the discrete version of exponential distribution, and because integral of an exponential function is still exponential, by the method in section 2, we expect the Gusein-Zade function to fit the ranked data well. It is again interesting that the best fitting function is not Gusein-Zade function, but Yule function.

The word frequency distribution is one of the most studied linguistic data, following the works by Zipf [1], and with tremendous amount of knowledge being accumulated on this topic [6]. Mandelbrot function is an example of extending one-parameter to two-parameter in fitting word frequency distribution [11]. Most usage of two-parameter models in linguistic data is perhaps on the phoneme distribution [10,29]. In our analysis, although several two-parameter models outperform the power-law function, the Zipf's law is still reasonably good.

In this paper we reveal some technical subtleties in fitting a function modified from a power-law function. Since power-law function is a linear function when both axes are in logarithmic scales, it is much easier to apply a linear regression on the log-transformed data. However, in the log-log plot, the density of data points at the low-ranking end (large r values) is much higher, even though their contribution to a visual impression is much less than their proportions. Whether or not we put a lesser weight on these points may affect the data fitting result.

Another issue not yet addressed in this paper is that a logarithmic transformation changes the data to another dataset, and the fitting result can be different from that before the logarithmic transformation. Since one major purpose of the logarithmic transformation is to apply an easier linear regression, in order to avoid a data transformation, non-linear regression is needed. Such analysis will be delayed to a future study.

Fitting linguistic data with more than two parameters (e.g., 3, 4) remains a possibility for future studies. However, for the three types of data considered here, the best R^2's are all larger than 0.95 using two-parameter functions, and there is no obvious systematic deviations. Two-parameter functions seem to achieve a good balance between under- and over-fittings. As in any statistical modeling, more does not always means better [45].

In conclusion, even though we have not exhausted all two-parameter functions in fitting ranked linguistic data, we at least put some of them together at one place that were used to be scattered in the literature. Some preliminary conclusions have been reached: among the functions we studied, and by fitting the log-transformed data, Beta function is the best fitting function for English letter frequency distribution, Yule is the best fitting function for inter-word spacing distribution, and Altmann function fits the word frequency distribution slightly better than others. However, these specific conclusions need confirmation in more datasets. It is important to emphasize the fact that a good statistical fitting could lead to a good phenomenological or first-principles approach to the same problem. We hope that this paper could contribute to pave the way in this direction for linguistic data.

Acknowledgments

We would like to thank Yaning Yang for discussions, and three anonymous reviewers for carefully reading the manuscript and providing many thoughtful comments.

References

1. Zipf, G.K. *The Psycho-Biology of Languages*; Houghton-Mifflin: Boston, MA, USA, 1935.
2. Herdan, G. *The Advanced Theory of Language as Choice and Chance*; Springer: Berlin, Germany, 1966.

3. Heaps, H.S. *Information Retrieval: Computational and Theoretical Aspects*; Academic Press: Orlando, FL, USA, 1978.
4. Menzerath, P. *Die Architektonik des Deutschen Wortschatzes*; Dümmler: Bonn, Germany, 1954.
5. Altmann, G. Prolegomena to Menzerath's law. *Glottometrika* **1980**, *2*, 1–10.
6. Baayen, R.H. *Word Frequency Distribution*; Kluwer Academic Publishers: Doredrecht, The Netherlands, 2001.
7. Mansilla, R.; Köppen, E.; Cocho, G.; Miramontes, P. On the behavior of journal impact factor rank-order distribution. *J. Informetrics* **2007**, *1*, 155–160.
8. Naumis, G.G.; Cocho, G. Tail universalities in rank distributions as an algebraic problem: The beta-like function. *Physica A* **2008**, *387*, 84–96.
9. Yule, G.U. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Phil. Tran. Royal Soc. London B* **1925**, *213*, 21–87.
10. Martindale, C.; Gusein-Zade, S.M.; McKenzie, D.; Borodovsky, M.Y. Comparison of equations describing the ranked frequency distributions of graphemes and phonemes. *J. Quant. Linguist.* **1996**, *3*, 106–112.
11. Mandelbrot, B. An information theory of the statistical structure of language. In *Proceedings of Symposium on Application Communication Theory, Butterworth*, London, UK, 22–26 September 1952; pp. 486–500.
12. Li, W. Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE Trans. Inf. Theory* **1992**, *38*, 1842–1845.
13. Gusein-Zade, S.M. On the frequency of meeting of key words and on other ranked series. *Sci. Tech. Inf. Ser. 2 Inf. Process. Syst.* **1987**, *1*, 28–32.
14. Weibull, W. A statistical distribution function of wide applicability. *J. Appl. Mech.* **1951**, *18*, 292–297.
15. Akaike, H. A new look at statistical model identification. *IEEE Trans. Automat. Control* **1974**, *19*, 716–722.
16. Venables, W.N.; Ripley, B.D. *Modern Applied Statistics with S-PLUS*, 2nd Edition; Springer-Verlap: New York, NY, USA, 1999.
17. David, H.A.; Nagaraja, H.N. *Order Statistics*, 3rd Edition; Wiley-Interscience: Malden, MA, USA, 2003.
18. Miller, G.A. Introduction. In *The Psycho-Biology of Languages*; MIT Press: Cambridge UK, 1965.
19. Rapoport A. Rank-size relations. In *International Encyclopedia of Statistics*; Kruskal, H., Tanur, J.M., Eds.; The Free Press: New York, NY, USA, 1978; Volume 2, pp. 847–854.
20. Kamecke, U. Testing the rank size rule hypothesis with an efficient estimator. *J. Urban Econ.* **1990**, *27*, 222–231.
21. Urzua, C.M. A simple and efficient test for Zipf's law. *Econ. Lett.* **2000**, *66*, 257–260.
22. Rousseau, R. George Kingsley Zipf: Life, ideas, his law and informatics. *Glottometrics* **2002**, *3*, 11–18.
23. Li, W. Zipf's law everywhere. *Glottometrics* **2002**, *5*, 14–21.

24. Evert, S. A simple LNRE model for random character sequences. In *Proceedings of Actes des 7es Journes Internationales d'Analyse Statistique des Donnes Textuelles (JADT)*, Louvain-la-Neuve, Belgium, 10–12 March 2004; pp. 411–422.
25. Lotka, A.J. The frequency distribution of scientific productivity. *J. Washington Acad. Sci.* **1926**, *16*, 317–323.
26. Kleinbaum, D.G.; Klein, M. *Survival Analysis: A Self-Learning Text*, 2nd Edition; Springer: New York, NY, USA, 2005.
27. Gusein-Zade, S.M. On the frequency distribution of letters in the Russian language. *Probl. Peredachi Informatsii* **1988**, *24*, 102–107 (in Russian).
28. Borodovsky, M.Y.; Gusein-Zade, S.M. A general rule for ranged series of codon frequencies in different genomes. *J. Biomol. Struct. Dyn.* **1989**, *6*, 1001–1012.
29. Nabeshima, T.; Gunji, T.P. Zipf's law in phenograms and Weibull distribution in ideograms: Comparison of English with Japanese. *BioSystems* **2004**, *73*, 131–139.
30. Altmann, E.G.; Pierrehumbert, J.B.; Motter, A.E. Beyond word frequency: Bursts, lulls, and scaling in the temporal distributions of words. *PLoS One* **2009**, *4*, e7678.
31. Shannon, C.E. Prediction and entropy of printed English. *Bell Sys. Tech. J.* **1951**, *29*, 50–64.
32. Richardson, M.; Gabrosek, J,; Reischman, D.; Curtiss, P. Morse code, scrabble, and the alphabet. *J. Stat. Educ.* **2004**, *12*, Avialable at: http://www.amstat.org/publications/jse/v12n3/richardson.html (accessed 6 April 2010).
33. Jernigan, R.W. A photographic view of cumulative distribution functions. *J. Stat. Educ.* **2008**, *16*, Avialable at: http://www.amstat.org/publications/jse/v16n1/jernigan.html (accessed 6 April 2010).
34. Martínez-Mekler, G.; Martínez, R.A.; Beltrán del Río, M.; Mansilla, R.; Miramontes, P.; Cocho, G. Universality of rank-ordering distributions in the arts and sciences. *PLoS One* **2009**, *4*, e4791.
35. Ortuño, M.; Carpena, P.; Bernaola-Galván, P.; Muñoz, E.; Somoza, A.M. Keyword detection in natural languages and DNA. *Phys. Rev. E* **2002**, *57*, 759–764.
36. Carpena, P.; Bernaola-Galván, P.; Hackenberg, M.; Coronado, A.V.; Oliver, J.L. Level statistics of words: Finding keywords in literary texts and symbolic sequences. *Europhys. Lett.* **2009**, *79*, 035102.
37. Li, W. Statistical properties of open reading frames in complete genome sequences. *Comput. Chem.* **1999**, *15*, 283–301.
38. McCoy, M.W.; Allen, A.P.; Gillooly, J.F. The random nature of genome architecture: Predicting open reading frame distributions. *PLoS One* **2009**, *4*, e6456.
39. Carpena, P.; Bernaola-Galván, P.; Román-Roldán, R.; Oliver, J.L. A simple and species-independent coding measure. *Gene* **2002**, *300*, 97–104.
40. Manaris, B.; Pellicoro, L.; Pothering, G.; Hodges, H. Investigating Esperanto's statistical proportions relative to other languages using neural networks and Zipf's law. In *Proceedings of the 2006 IASTED International Conference on Artificial Intelligence and Applications (AIA 2006)*, Innsbruck, Austria, 13–16 February 2006; ACTA Press: Anaheim, CA, USA, 2006; pp. 102–108.
41. Davis M. *A Frequency Dictionary of Spanish: Core Vocabulary for Leaners*; Routledge: New York, NY, USA, 2006.

42. Ferrer i Cancho, R.; Sole, R.V. Two regimes in the frequency of words and the origins of complex lexicons: Zipf's law revisited. *J. Quant. Linguist.* **2001**, *8*, 165–173.
43. Ha, L.Q.; Sicilia-Garcia, E.I.; Maing, J.; Smith, F.J. Extension of Zipf's law to word and character n-grams for English and Chinese. *J. Comp. Linguist. Chin. Lang. Process.* **2003**, *8*, 77–102.
44. Li, W.; Freudenberg, J. Two-parameter characterization of chromosome-scale recombination rate. *Genome Res.* **2009**, *19*, 2300–2307.
45. Li, W. The-more-the-better and the-less-the-better. *Bioinformatics* **2006**, *22*, 2187–2188.
46. Baayen, R.H. *Analyzing Linguistic Data—A Practical Introduction to Statistics Using R*; Cambridge University Press: Cambridge, UK, 2008.
47. Evert, S.; Baroni, M. zipfR: Word frequency distributions in R. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, 25-27 June 2007; Association for Computational Linguistics: Morristown, NJ, USA, 2007; pp. 29–32.
48. Ryan, T.P. *Modern Regression Methods*; John Wiley & Sons: New York, NY, USA, 1997.

Appendix A: Source of Texts

The text of Herman Melville's *Moby-Dick* was downloaded from the Project Gutenberg website (http://www.gutenberg.org/). A few pre-processing steps have been carried out, such as removing the chapter titles, removing punctuations, manually transform some colloquial terms to their corresponding formal terms (I'd $\rightarrow$ "I would", ne'er $\rightarrow$ never, can't $\rightarrow$ "can not", 'tis $\rightarrow$ "it is", *etc.*), and converting uppercases to lowercases.

The text contains more than 212,000 tokens with 16713 word types. 7267 word types (43% of all) appear only once, also known as *hapax legomenon*. Similarly, 60%, 70%, and 75% of all word types appear at most 2, 3, and 4 times.

The Spanish text of *Don Quijote* was also downloaded from the Project Gutenberg website, with more than 80,000 tokens and more than 23000 word types. Around half (49%) of the word types ($\sim$ 11000) are hapax legomenon.

Appendix B: Using R Statistical Package:

We use the statistical package R (http://www.r-project.org/) (for linguistic applications using R, see [46,47]). The regression $y = a_0 + a_1 x_a + a_2 x_2$ is represented by a shorthand notation in R: $y \sim x_1 + x_2$. A typical R session for the above multiple regression is (the texts after # are comments):

```
lm(y ~ x1 + x2)      # for "linear model"
res = lm(y ~ x1 + x2)   # save the result to an object
summary(res)         # details of the result
res$coefficients     # regression coefficients
```

The R command for regression with weights (suppose w is an array of weights) is:

```
lm(y ~ x , weights=w).
```

Appendix C: The Condition for the Equivalence of Two Definitions of R^2 in Linear Regression

The concept of coefficient of determination R^2 is based on the variance decomposition. Suppose our raw data is $\{x_i, y_i\}(i = 1, 2, \cdots n)$, the following variance decomposition holds true for the regression models for $y \sim x$:

$$\sum_i (y_i - \overline{y})^2 = \sum_i (y_i - \hat{y}_i)^2 + \sum_i (\hat{y}_i - \overline{y})^2 = SSE + \sum_i (\hat{y}_i - \overline{y})^2 \tag{22}$$

where $\overline{y}$ for the mean of y, $\hat{y}$ for the fitted value by regression model, and the name SSE means sum of squared error. For the variance decomposition to be true, the cross-product term should be zero. The cross-product term in Equation (22) is $2\sum_i (y_i - \hat{y}_i)(\hat{y}_i - \overline{y})$. It is indeed zero because $\sum_i (y_i - \hat{y}_i) = 0$ and $\sum_i (y_i - \hat{y}_i)\hat{y}_i = 0$ conditions have to be satisfied as these are related to the derivative with respect to the two coefficients of the sum-of-square-error. For the sum-of-square-error to reach the minimum, the derivatives have to be zero.

Similarly variance decomposition can be obtained for the $\log(y) \sim \log(x)$ regression:

$$\sum_i (\log(y_i) - \overline{\log(y)})^2 = \sum_i (\log(y_i) - \hat{\log(y_i)})^2 + \sum_i (\hat{\log(y_i)} - \overline{\log(y)})^2 \tag{23}$$

When variance decomposition holds, ratio of a variance component to the total then can be defined, for $y \sim x$ regression:

$$R^2 = \frac{\sum_i (\hat{y}_i - \overline{y})^2}{\sum_i (y_i - \overline{y})^2} = 1 - \frac{SSE}{\sum_i (y_i - \overline{y})^2} \tag{24}$$

and for $\log(y) \sim \log(x)$ regression:

$$R^2_{log} = \frac{\sum_i (\hat{\log(y_i)} - \overline{\log(y)})^2}{\sum_i (\log(y_i) - \overline{\log(y)})^2} \tag{25}$$

Note that R^2_{log} is generally not identical to R^2 as the regression is applied to the transformed data.

There is another definition of R^2 based on the correlation coefficient between the observed and the fitted y value:

$$\begin{aligned} R^2_{corr} &= Cor(y, \hat{y})^2 = \frac{|\sum_i (y_i - \overline{y})(\hat{y}_i - \overline{y})|^2}{\sum_i (y_i - \overline{y})^2 \sum_i (\hat{y}_i - \overline{y})^2} \\ R^2_{corr,log} &= Cor(\log(y), \log(\hat{y}))^2 \end{aligned} \tag{26}$$

For the regression $y \sim x$,

$$\begin{aligned} & \sum_i (y_i - \overline{y})(\hat{y}_i - \overline{y}) = \sum_i (y_i - \hat{y}_i + \hat{y}_i - \overline{y})(\hat{y}_i - \overline{y}) \\ = & \sum_i (y_i - \hat{y}_i)(\hat{y}_i - \overline{y}) + \sum_i (\hat{y}_i - \overline{y})^2 \\ = & \sum_i (y_i - \hat{y}_i)\hat{y}_i - \overline{y}\sum_i (y_i - \hat{y}_i) + \sum_i (\hat{y}_i - \overline{y})^2 \\ = & \sum_i (\hat{y}_i - \overline{y})^2 \end{aligned} \tag{27}$$

making R^2_{corr} in Equation (26) equal to R^2 in Equation (24).

When variance decomposition does not hold, such as the case when the parameter values are not estimated by least square, the two definitions of R^2 are not equivalent. In that situation, R^2_{corr} seems to be more robust (see R^2_{raw} on p.194 of [48]). It is always recommended to explicitly state which definition of R^2 is used (p.79 of [48]).

Appendix D: Weighted Linear Regression

The standard regression $y \sim x$ by least square fitting aims at minimize the expression $\sum_i (y_i - a - bx_i)^2$. In order to assign priority to data points in their contribution to the regression, it is possible to minimize $\sum_i w_i(y_i - a - bx_i)^2$, where $\{w_i\}(i = 1, 2, \cdots n)$ are pre-set weightings. For example, for ranked distributions, one may decide that the low-rank-number points are more important than low-ranking points. When one is more interested in fitting low-rank-number points than low-ranking points, it is possible to choose a monotonically decreasing weight function, such as $w_i = 1/i$.

It can be shown that variance decomposition continues to be true:

$$\sum_i w_i(y_i - \langle y\rangle_w)^2 = \sum_i w_i(y_i - \hat{y}_i)^2 + \sum_i w_i(\hat{y}_i - \langle y\rangle_w)^2 = SSE_w + \sum_i w_i(\hat{y}_i - \langle y\rangle_w)^2 \qquad (28)$$

where $\langle y\rangle_w$ for the weighted mean of y: $\langle y\rangle_w \equiv \sum_i w_i y_i / \sum_i w_i$. Note that the weighted mean $\langle y\rangle_w$ can not be replaced by the unweighted mean $\overline{y}$.

Similar variance decomposition for the log-scaled regression is:

$$\sum_i w_i(\log(y_i) - \langle \log(y)\rangle_w)^2 = \sum_i w_i(\log(y_i) - \log\hat{(}y_i))^2 + \sum_i w_i(\log\hat{(}y_i) - \langle \log(y)\rangle_w)^2 \qquad (29)$$

With the variance decomposition, R^2 can be defined for the weighted regression:

$$\begin{aligned} R^2_{weight} &= \frac{\sum_i w_i(\hat{y}_i - \langle y\rangle_w)^2}{\sum_i w_i(y_i - \langle y\rangle_w)^2} = 1 - \frac{SSE_w}{\sum_i w_i(y_i - \langle y\rangle_w)^2} \\ R^2_{log,weight} &= \frac{\sum_i w_i(\log\hat{(}y_i) - \langle \log(y)\rangle_w)^2}{\sum_i w_i(\log(y_i) - \langle \log(y)\rangle_w)^2} \end{aligned} \qquad (30)$$

The correlation-based definition of R^2 for weighted regression is:

$$\begin{aligned} R^2_{corr,weight} &= \frac{|\sum_i w_i(y_i - \langle y\rangle_w)(\hat{y}_i - \langle y\rangle_w)|^2}{\sum_i w_i(y_i - \langle y\rangle_w)^2 \cdot \sum_i w_i(\hat{y}_i - \langle y\rangle_w)^2} \\ R^2_{corr,log,weight} &= \frac{|\sum_i w_i(\log(y_i) - \langle \log(y)\rangle_w)(\log\hat{(}y_i) - \langle \log(y)\rangle_w)|^2}{\sum_i w_i(\log(y_i) - \langle \log(y)\rangle_w)^2 \cdot \sum_i w_i(\log\hat{(}y_i) - \langle \log(y)\rangle_w)^2} \end{aligned} \qquad (31)$$

We have so far defined eight R^2's: based on variance ratio or based on correlation square, based on regression on log scale or linear scale, with or without weight. Some of them are equivalent under certain conditions, others are not.

Appendix E: Relationship between AIC and SSE in Weighted Regression

We examine whether the relationship between AIC and SSE discussed in Section 1 is still true for weighted regression. The most important assumption made is that the log-likelihood is weighted, not likelihood itself, and under the normal noise model:

$$\log(L) = \sum_i^n w_i \log(L_i) = \sum_i^n w_i \log \frac{e^{-\frac{(y_i - \hat{y}_i(^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} = -\frac{n_w}{2}\log(2\pi) - \frac{n_w}{2}\log(\sigma^2) - \frac{\sum_i w_i (y_i - \hat{y}_i)^2}{2\sigma^2} \quad (32)$$

where $n_w = \sum_i w_i$. When the variance is unknown, it can be estimated as $\hat{\sigma^2} = \sum_i w_i (y_i - \hat{y}_i)^2 / \sum_i w_i = SSE_w / n_w$. Plugging the estimated variance in the above equation:

$$\log(\hat{L}) = C - \frac{n_w}{2} \log \frac{SSE_w}{n_w} \quad (33)$$

so AIC can be derived (after removing constant terms):

$$AIC = -2\log(\hat{L}) + 2K = n_w \log \frac{SSE_w}{n_w} + 2K \quad (34)$$