Upper Bound of Entropy Rate Revisited —A New Extrapolation of Compressed Large-Scale Corpora—

Ryosuke Takahira[1] **Kumiko Tanaka-Ishii**[2] **Łukasz Dębowski**[3]
[1]Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan
takahira@limu.ait.kyushu-u.ac.jp
[2]Research Center for Advanced Science and Technology, University of Tokyo, Japan
kumiko@cl.rcast.u-tokyo.ac.jp
[3]Institute of Computer Science, Polish Academy of Sciences, Poland
ldebowsk@ipipan.waw.pl

Abstract

The article presents results of entropy rate estimation for human languages across six languages by using large, state-of-the-art corpora of up to 7.8 gigabytes. To obtain the estimates for data length tending to infinity, we use an extrapolation function given by an ansatz. Whereas some ansatzes of this kind were proposed in previous research papers, here we introduce a stretched exponential extrapolation function that has a smaller error of fit. In this way, we uncover a possibility that the entropy rates of human languages are positive but 20% smaller than previously reported.

1 Introduction

Estimation of the entropy rate of natural language is a challenge originally set up by Shannon (Shannon, 1948; Shannon, 1951). The entropy rate quantifies the complexity of language, precisely the rate how fast the amount of information grows in our communication with respect to the text length. Today, the entropy rate provides an important target for data compression algorithms, where the speed of convergence of the compression rate to the entropy rate is an informative benchmark. Measuring the entropy rate is also the first step in answering what kind of a stochastic process can model generation of texts in natural language, an important question for many practical tasks of natural language engineering.

An important theoretical question concerning the entropy rate, which has also been noted in the domains of computational linguistics (Genzel and Charniak, 2002) and speech processing (Levy and Jaeger, 2007), is whether the entropy rate of human language is a strictly positive constant. The overwhelming evidence collected so far suggests that it is so—in particular, the amount of information communicated per unit time in English text is generally agreed to be about 1 bpc (bit per character) (Shannon, 1951; Cover and King, 1978; Brown et al., 1983; Schümann and Grassberger, 1996). Although this is what we might intuitively expect, Hilberg formulated a hypothesis that the entropy rate of natural language is zero (Hilberg, 1990). Zero entropy rate does not imply that the amount of information in texts is not growing, but that it grows with a speed slower than linear. From this perspective we want to provide as exact estimates of the entropy rate for natural language as possible.

Precise estimation of the entropy rate is a challenging task mainly because, mathematically speaking, the sought parameter is a limit for text length tending to infinity. To alleviate this problem, previous great minds proposed estimation methods based on human cognitive testing (Shannon, 1951; Cover and King, 1978). Since human testing is costly, however, such attempts remain limited in terms of the scale and number of tested languages. In contrast, although any conceivable data size can only be finite, today's language data have become so large in scale that we may reconsider estimation of the entropy rate using big data computation. This point was already raised by (Shannon, 1948), which led to important previous works such as (Brown et al., 1983) in the domain of computational linguistics. Both of these articles and many other that followed, however, mostly considered the English language only.

In contrast, in this article, we present the results of entropy rate estimation using state-of-the-art large data sets in six different languages, including up to 7.8 gigabytes of data in English. We try to estimate the entropy rate by compressing these data sets using the PPM algorithm and extrapolating the data points with a carefully selected ansatz function. Whereas a couple of ansatz functions were previously

Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity,
pages 213–221, Osaka, Japan, December 11-17 2016.

proposed in (Hilberg, 1990; Crutchfield and Feldman, 2003; Ebeling and Nicolis, 1991; Schümann and Grassberger, 1996), here we introduce another function, which is a stretched exponential function and enjoys the same number of parameters as previous proposals. The new functions yields a smaller error of fit. As a result, we arrive at the entropy rate estimates which are positive but 20% smaller than previously reported.

2 Entropy Rate

Let X_1^∞ be a stochastic process, i.e., an infinite sequence of random variables $X = X_1, X_2, X_3, \ldots$ with each random variable X_i assuming values $x \in \mathbb{X}$, where $\mathbb{X}$ is a certain set of countably many symbols. For natural language, for instance, $\mathbb{X}$ can be a set of characters, whereas X_1^∞ is an infinite corpus of texts. Let X_i^j, where $i \leq j$, denote a finite subsequence $X_i^j = X_i, X_{i+1}, \ldots, X_j$ of X_1^∞ and let $P(X_i^j = x_i^j)$ denote a probability function of the subsequence X_i^j. The Shannon entropy of a finite subsequence X_i^j is defined as:

$$H(X_i^j) = -\sum_{x_i^j} P(X_i^j = x_i^j) \log_2 P(X_i^j = x_i^j), \tag{1}$$

where sequences x_i^j are instances of X_i^j (Shannon, 1948). In contrast, the entropy rate of the infinite sequence X is defined as (Cover and Thomas, 2006):

$$h = \lim_{n\to\infty} \frac{H(X_1^n)}{n}. \tag{2}$$

The entropy rate is the amount of information per element for the data length tending to infinity.

Let us note that the entropy rate quantifies the asymptotic growth of the number of possible values of an infinite sequence X_1^∞. Roughly speaking, there are effectively only 2^{nh} possible values for a subsequence X_1^n, where n is the sequence length. In other words, condition $h > 0$ is tantamount to an exponential growth of the number of possible sequences with respect to n. Value $h = 0$ need not mean that the number of possibilities does not grow. For instance, for a sequence X_1^n whose number of possibilities grows like $2^{A\sqrt{n}}$, as supposed by Hilberg (1990), we have $h = 0$. Although the number of possibilities for such a sequence of random variables grows quite fast, the speed of the growth cannot be properly measured by the entropy rate.

The entropy rate thus quantifies, to some extent, the degree of randomness or freedom underlying the text characters to follow one another. For human languages, the occurrence of a linguistic element, such as a word or character, depends on the previous elements, and there are many long repetitions. This results in a lower value of the entropy rate than for a random sequence, but the ultimate degree of randomness in natural language is hard to simply guess. Whereas Hilberg (1990) supposed that $h = 0$ holds for natural language, this is only a minority view. According to the overwhelming experimental evidence the entropy of natural language is strictly positive (Shannon, 1951; Cover and King, 1978; Brown et al., 1983; Schümann and Grassberger, 1996). We may ask however whether these known estimates are credible. In fact, if convergence of $H(X_1^n)/n$ to the entropy rate is very slow, this need not be so. For this reason, while estimating the entropy rate, it is important to investigate the speed of the estimate convergence.

3 Direct estimation methods

There are several methods to estimate the entropy rate of natural language. These can be largely divided into methods based on human cognitive testing and methods based on machine computation. Estimation via human cognitive testing is mainly conducted by showing a substring of a text to a human examinee and having him or her guess the character to follow the substring. This method was introduced by Shannon (1951). He tested an unmentioned number of examinees with the text of Dumas Malone's "Jefferson

the Virginian" and obtained $h \approx 1.3$ bpc. This method was improved by Cover and King (1978) as a sort of gambling. The results with 12 examinees produced an average of $h \approx 1.34$ bpc. Human cognitive testing has the advantage over methods based on machine computations that the estimates of entropy rate converge faster. Unfortunately, such human cognitive testing is costly, so the number of examinees involved is small and the samples are rather short. It is also unclear whether human examinees guess the text characters according to the true probability distribution.

In contrast, today, estimation of the entropy rate can be performed by big data computation. For this paradigm, two specific approaches have been considered so far.

1. The first approach is to estimate the probabilistic language models underlying formula (2). A representative classic work is (Brown et al., 1983), who reported $h \approx 1.75$ bpc, by estimating the probability of trigrams in the Brown National Corpus.

2. The second approach is to compress the text using a data compression algorithm. Let $R(X_1^n)$ denote the size in bits of text X_1^n *after* the compression. Then the code length per unit, $r(n) = R(X_1^n)/n$, is always larger than the entropy rate (Cover and Thomas, 2006),

$$r(n) \geq h. \tag{3}$$

   We call $r(n)$ the *encoding rate* in the rest of this article.

In the following we will apply the second approach. In fact, there are various algorithms to compress texts. Within our context we are interested in *universal* methods. A universal text compressor guarantees that the encoding rate converges to the entropy rate, provided that the stochastic process X_1^∞ is stationary and ergodic, i.e., equality

$$\lim_{n\to\infty} r(n) = h \tag{4}$$

holds with probability 1. Among the important known universal compressors we can name: the Lempel-Ziv (LZ) code (Ziv and Lempel, 1977), the PPM code (Bell et al., 1990), and a wide class of grammar-based codes (Kieffer and Yang, 2000), with many particular instances such as SEQUITUR (Nevill-Manning and Witten, 1997) and NSRPS (Non-Sequential Recursive Pair Substitution) (Ebeling and Nicolis, 1991; Grassberger, 2002). Whereas all these codes are universal, they are not equal. Let us briefly describe some properties of these compressors. First of all, they are based on different principles. The LZ code and the grammar-based codes compress texts roughly by detecting repeated substrings and replacing them with shorter identifiers. A proof of universality of the LZ code can be found in (Cover and Thomas, 2006), whereas the proof of universality of grammar-based codes can be found in (Kieffer and Yang, 2000). In contrast, the PPM code is an n-gram based language modeling method (Bell et al., 1990) which applies variable length n-grams and arithmetic coding. The PPM code is guaranteed to be universal when the length of the n-gram is considered up to infinity (Ryabko, 2010).

A very important question for our application is the scaling of the encoding rate of universal codes for finite real data. Since the probabilistic model of natural language remains unknown, the notion of universality may serve only as a possible standard to obtain a stringent upper bound. One may raise some doubt that natural language is strictly stationary since the word probabilities do vary across time, as indicated by (Baayen, 2001). Moreover, many off-the-shelf compressors are not strictly universal, since they are truncated in various ways to gain the computational speed. Therefore, a suitable compressor can only be chosen through experimental inspection.

Among state-of-the-art compressors, we have considered zip, lzh, tar.xz, and 7-zip LZMA for the LZ methods and 7-zip PPMd for the PPM code. In Figure 1 (right panel) we show how the encoding rate depends on the data length for a Bernoulli process with $p = 0.5$ (left panel, listed later in the first line of the third block of Table 1) and for natural language data of Wall Street Journal corpus (right panel, listed in the third line of the third block of Table 1). First, let us consider the Bernoulli process, which is a simple artificial source. Formally, the Bernoulli process is a sequence of independent random variables taking the value of 1 with probability p and 0 with probability $1 - p$. There are two known theoretical results for this process: The theoretically proven encoding rate of the LZ code is as much as

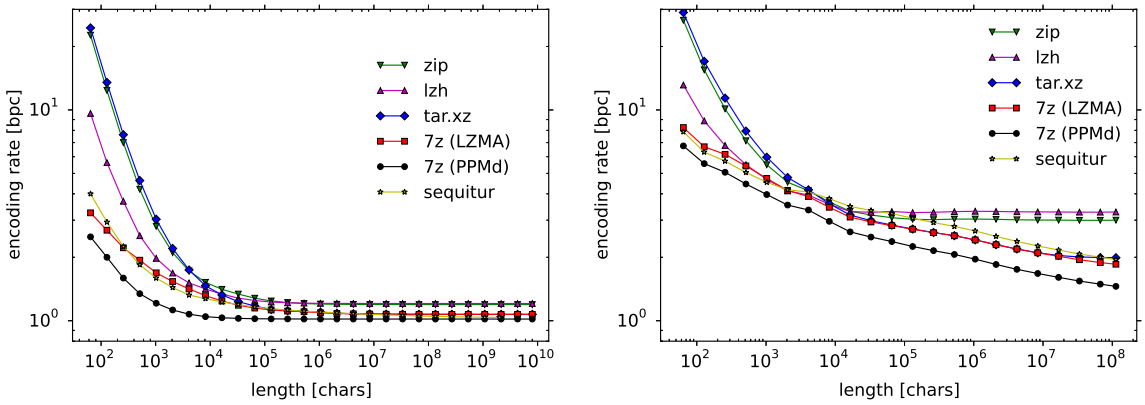

Figure 1: Compression results for a Bernoulli process ($p = 0.5$, left panel) and Wall Street Journal (right panel) for LZ, PPM, and SEQUITUR.

$r(n) = A/(\log n) + h$ (Louchard and Szpankowski, 1997), whereas the encoding rate for the PPM code is proved to be only $r(n) = A(\log n)/n + h$ (Barron et al., 1998; Atteson, 1999). Thus the convergence is extremely slow for the LZ code and quite fast for the PPM code. This exactly can be seen in Figure 1 (left panel), where all data points for the LZ code remain way above 1.0 bpc, the true entropy rate, while the data points for the PPM code practically converge to 1.0 bpc.

As for natural language data, whereas the empirical speed of convergence is much slower for the Wall Street Journal, the gradation of the compression algorithms remains the same. Algorithms such as zip and lzh get saturated probably because they are truncated in some way, whereas SEQUITUR, 7-zip LZMA and 7-zip PPMd gradually improve their compression rate the more data they read in. Since the encoding rate is visibly the smallest for 7-zip PPMd, in the following, we will use this compressor to estimate the entropy rate for other natural language data.

4 Extrapolation Functions

Many have attempted to estimate the entropy rate via compression. For example, paper (Bell et al., 1990) reported $h \approx 1.45$ bpc for the collected works of Shakespeare in English. Majority of the previous works, however, reported only a single value of the encoding rate for the maximal size of the available data. Whereas any computation can handle only a finite amount of data, the true entropy rate is defined in formula (2) as a limit for infinite data. The later fact should be somehow taken into consideration, especially if convergence (4) is slow, which is the case of natural language. One way to fill this gap between the finite data and the infinite limit is to use extrapolation. In other words, the encoding rate $r(n)$ is calculated for many n and the plots are extrapolated using some function $f(n)$. Since the probabilistic model of natural language is unknown, function $f(n)$ has been considered so far in form of an ansatz.

Previously, two ansatzes have been proposed, to the best of our knowledge. The first one was proposed by Hilberg (1990). He examined the original paper of (Shannon, 1951), which gives a plot of some upper bounds of $H(X_1^n)/n$. Since Hilberg believed that the entropy rate vanishes, $h = 0$, his ansatz was

$$f_0(n) = An^{\beta-1}, \tag{5}$$

with $\beta \approx 0.5$, according to Hilberg. If we do not believe in a vanishing entropy rate, the above formula can be easily modified as

$$f_1(n) = An^{\beta-1} + h, \tag{6}$$

so that it converges to an arbitrary value of the entropy rate, cf., (Crutchfield and Feldman, 2003). Another ansatz was given in papers (Ebeling and Nicolis, 1991) and (Schümann and Grassberger, 1996). It reads

$$f_2(n) = An^{\beta-1} \ln n + h. \tag{7}$$

Table 1: Data used in this work, its size, its encoding rate, entropy rate and the error

Text	Language	Size (chars)	encoding rate (bit)	$f_1(n)$ h (bit)	$f_1(n)$ error $\times 10^{-2}$	$f_3(n)$ h (bit)	$f_3(n)$ error $\times 10^{-2}$
Large Scale Random Document Data							
Agence France-Presse	English	4096003895	1.402	1.249	1.078	1.033	0.757
Associated Press Worldstream	English	6524279444	1.439	1.311	1.485	1.128	1.070
Los Angeles Times/Washington Post	English	1545238421	1.572	1.481	1.108	1.301	0.622
New York Times	English	7827873832	1.599	1.500	0.961	1.342	0.616
Washington Post/Bloomberg	English	97411747	1.535	1.389	1.429	1.121	0.991
Xinhua News Agency	English	1929885224	1.317	1.158	0.906	0.919	0.619
Wall Street Journal	English	112868008	1.456	1.320	1.301	1.061	0.812
Central News Agency of Taiwan	Chinese	678182152	5.053	4.459	1.055	3.833	0.888
Xinhua News Agency of Beijing	Chinese	383836212	4.725	3.810	0.751	2.924	0.545
People's Daily (1991-95)	Chinese	101507796	4.927	3.805	0.413	2.722	0.188
Mainichi	Japanese	847606070	3.947	3.339	0.571	2.634	0.451
Le Monde	French	727348826	1.489	1.323	1.103	1.075	0.711
KAIST Raw Corpus	Korean	130873485	3.670	3.661	0.827	3.327	1.158
Mainichi (Romanized)	Japanese	1916108161	1.766	1.620	2.372	1.476	2.067
People's Daily (pinyin)	Chinese	247551301	1.850	1.857	1.651	1.667	1.136
Small Scale Data							
Ulysses (by James Joyce)	English	1510885	2.271	2.155	0.811	1.947	1.104
À la recherche du temps perdu (by Marcel Proust)	French	7255271	1.660	1.414	0.770	1.078	0.506
The Brothers Karamazov (by Fyodor Dostoyevskiy)	Russian	1824096	2.223	1.983	0.566	1.598	0.839
Daibosatsu toge (by Nakazato Kaizan)	Japanese	4548008	4.296	3.503	1.006	2.630	0.875
Dang Kou Zhi (by by Wan-Chun Yu)	Chinese	665591	6.739	4.479	1.344	2.988	1.335

Using this ansatz, paper (Schümann and Grassberger, 1996) obtained $h \approx 1.7$ bpc for the collected works of Shakespeare and $h \approx 1.25$ bpc for the LOB corpus of English.

We have used up to 7.8 gigabytes of data for six different languages and quite many plots were available for fitting, as compared to previous works. As will be shown in §6.1, function $f_1(n)$ does not fit well to our plots. Function $f_1(n)$, however, is no more than *some* ansatz. If we can devise another ansatz that fits better, then this should rather be used to estimate the entropy rate. In fact we have come across a better ansatz. The function we consider in this article is a stretched exponential function,

$$f_3(n) = \exp(An^{\beta-1} + h'), \tag{8}$$

which embeds function $f_1(n)$ in an exponential function and yields the entropy rate $h = \exp h'$. In fact, function $f_3(n)$ converges to h slower than $f_1(n)$. In a way, this is desirable since slow convergence of the encoding rate is some general tendency of the natural language data. As a by-product, using function $f_3(n)$ we will obtain smaller estimates of the entropy rate than using function $f_1(n)$.

5 Experimental Procedure

5.1 Data preparation

Table 1 lists our data, including each text, its language and size in the number of characters, its encoding rate using the full data set (the minimal observed encoding rate), and the extrapolation results for the entropy rate h, including the error of the estimates—as defined in §5.2 and analyzed later. We carefully chose our data by examining the redundancies. Many of the freely available large-scale corpora suffer from poor quality. In particular, they often contain artificially long repetitions. Since such repetitions affect the entropy rate estimates, we have only used corpora of a carefully checked quality, making sure that they do not contain large chunks of a repeated text.

The table contains two blocks. The first block contains state-of-the-art large-scale corpora of texts. As will be shown in our experiments, the plots for the raw corpora often oscillated due to the topic change. To overcome this problem we have performed randomization and averaging. First, we have shuffled the corpora at the level of documents and, second, we have averaged ten different random permutations for

each corpus. The experimental results shown from the 4th column to the last one of Table 1 pertain to so processed language data. As for the Japanese and Chinese data, in addition to the original texts of the Mainichi and People's Daily newspapers, the Romanized versions were generated.[1] In contrast, the second block of Table 1 contains long literary works in five different languages. These data have not been randomized. The data in the first and second blocks encompass six different languages.

5.2 Detailed procedure

To estimate the entropy rate, we have used the 7-zip compressor, which implements the PPMd algorithm. As discussed in §3, this compressor seems the best among state-of-the-art methods. It compresses best not only the real Wall Street Journal corpus but also the artificial Bernoulli process. For this reason, we have used this compressor. Further detailed options of the PPMd algorithm were carefully chosen. Since the 7-zip program compresses by recording statistics for file names as well, the input text was fed to the compressor via a Unix pipe so that the compression was conducted *without* a file name. We also carefully excluded the *header* of the compressed file (which includes the name of the compressor etc.). This header is included in the compressed file but does not count to the proper compression length.

Another important option of the 7-zip program concerns the maximal n-gram length used by the PPM, called here MAX. As noted in §3, when MAX is infinite the compression method is universal. But the larger MAX is, the slower the compression procedure becomes. Therefore, any available compressor sets an upper bound on MAX, whereas the user can choose the MAX value smaller than this bound (the bound equals 32 in the case of 7-zip PPMd). However, even within this preset range, it was not always the case that a larger MAX resulted in a better encoding rate. Therefore, in our work, for each full data set, we searched for the value of MAX that achieved the best encoding rate and consistently used those best encoding rates for different subsets of the full data set.

Having clarified these specific issues, our detailed experimental procedure, applied to each data set from Table 1, was as follows. First, for every $n = 2^k$, where $k = 6, 7, \ldots, \log_2(\text{data size})$, the first n characters of the full text were taken. This subsequence, denoted X_1^n, was then compressed using the 7-zip program, and its size $R(X_1^n)$ in bits was measured to calculate the encoding rate $r(n) = R(X_1^n)/n$. The obtained encoding rates for different n were fitted to the ansatz functions $f(n) = f_j(n)$, where $j = 1, 2, 3, 4$. When encoding rates $r(n_i) = R(X_1^{n_i})/n_i$ for K distinct values of n_i were obtained, the fit was conducted by minimizing the square error as follows:

$$error = \sqrt{\frac{\sum_{i=1}^{K} (\ln r(n_i) - \ln f(n_i))^2}{K}}. \tag{9}$$

The logarithm was taken here to ascribe a larger weight to the errors of the larger n, since we were particularly interested in the tail behavior of the data points.

6 Experimental Results

6.1 Fitting Results

Figure 2 shows our results for the Wall Street Journal (WSJ) corpus (Table 1, first block, seventh line), which is the benchmark corpus most typically used in the computational processing of human language. The figure shows the encoding rate $r(n)$ (vertical axis) as a function of the text size in characters n (horizontal axis). The left panel of Figure 2 shows the results obtained from the original text. The encoding rates tend to oscillate, which is due to topic changes in the corpus. Such oscillation is visible in majority of the natural language data, where some data can oscillate much worse than WSJ. In the context of entropy rate estimation such oscillation was already reported in paper (Schümann and Grassberger, 1996). Some possible way to cope with this problem is to shuffle the text at the level of documents. The right panel of Figure 2 shows the average encoding rate for the data 10-fold shuffled by documents. The data points in the right panel oscillate less than in the left panel. At the same time, since shuffling the documents introduces some randomness, the entropy rate estimate is about 1% larger for the randomized data

[1] KAKASI and Pinyin Python library software were used to Romanize Japanese and Chinese, respectively.

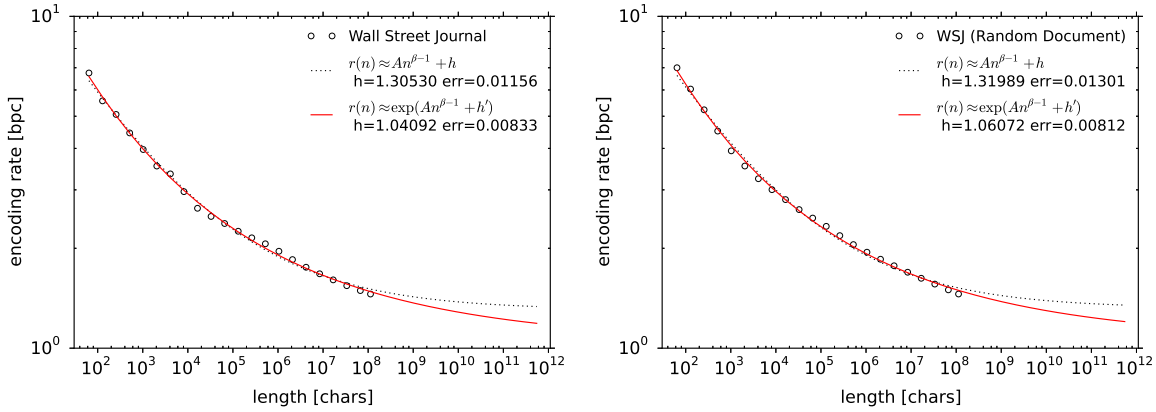

Figure 2: Encoding rates for the Wall Street Journal corpus (in English). The left panel is for the original data, whereas the right panel is the average of the data 10-fold shuffled by documents. To these results we fit functions $f_1(n)$ and $f_3(n)$.

than for the original corpus. Both panels of Figure 2 show two fits of the encoding rate, to extrapolation functions $f_1(n)$ and $f_3(n)$—given by formulae (6) and (8), respectively. Whereas, visually, it is difficult to say which of the functions fits better, we can decide on that using the value of error (9). The estimates of the entropy rate are $h = 1.32$ with $error$ being 0.0130 for $f_1(n)$ and $h = 1.061$ with $error$ being 0.00812 for $f_3(n)$. We can suppose that function $f_3(n)$ yields both a smaller entropy rate estimate and a smaller fitting error.

This hypothesis can be confirmed. We conducted the analogous fitting to all our data sets for three ansatz functions $f_1(n)$, $f_2(n)$, and $f_3(n)$. The fitted values of h and $error$ for $f_1(n)$ and $f_3(n)$, for both 10-fold randomized corpora and non-randomized texts are listed in Table 1 in the last four columns. The average values of the $error$ for $f_1(n)$, $f_2(n)$ and $f_3(n)$ were 0.0113, 0.0194, and 0.00842 across all data sets, respectively. The plots therefore fit the best to $f_3(n)$. Among the three ansatz functions, function $f_2(n)$ is the worst choice. In contrast, the stretched exponential function $f_3(n)$ seems better than the modified Hilberg function $f_1(n)$ and it consistently yields smaller estimates of the entropy rate.

6.2 A Linear Perspective

If the exponent β does not depend on a particular corpus of texts, i.e., if it is some language universal, then for all three functions $f_1(n)$, $f_2(n)$, and $f_3(n)$ we can draw a diagnostic linear plot with axes: $Y = r(n)$ and $X = n^{\beta-1}$ for $f_1(n)$, $Y = r(n)$ and $X = n^{\beta-1} \ln n$ for $f_2(n)$, and $Y = \ln r(n)$ and $X = n^{\beta-1}$ for $f_3(n)$, respectively. In these diagnostic plots, the entropy rate corresponds to the intercept of the straight line on which the data points lie approximately. Since we observe that exponent β is indeed some language universal, we use these plots to compare different text corpora.

In these plots, ansatzes $f_1(n)$, $f_2(n)$, and $f_3(n)$ can be analyzed as a form of linear regression. Let us focus on $f_3(n)$, the function that yields the minimal fitting error. If we put $Y = \ln r(n)$ as the vertical axis and $X = n^{\beta-1}$ as the horizontal axis where $\beta = 0.884$, the average value for the fit to $f_3(n)$, then the plots for all large scale natural language data (first block of Table 1) can be transformed as shown in Figure 3. It can be seen that each set of data points is roughly assembled in a linear manner.

In Figure 3, the black points are English, the white ones are Chinese, and the gray ones are other languages including Romanized Chinese and Japanese. Two main groups of plots can be seen in Figure 3, one lower and one upper, where the lower plots in black are for English and the upper plots in white are for Chinese . The results for other languages, shown in gray, are located somewhere between English and Chinese. The gray plots appearing amidst the lower group indicate Romanized Japanese and Chinese. These results show that the script type distinguishes the amount of information per character.

Two straight lines were obtained in Figure 3 for the English and Chinese groups by least squares fitting to all data points from each group, respectively. Since the horizontal axis indicates variable $X = n^{\beta-1}$, condition $n \to \infty$ corresponds to condition $X = 0$. The intercept of a fitted straight line is thus the

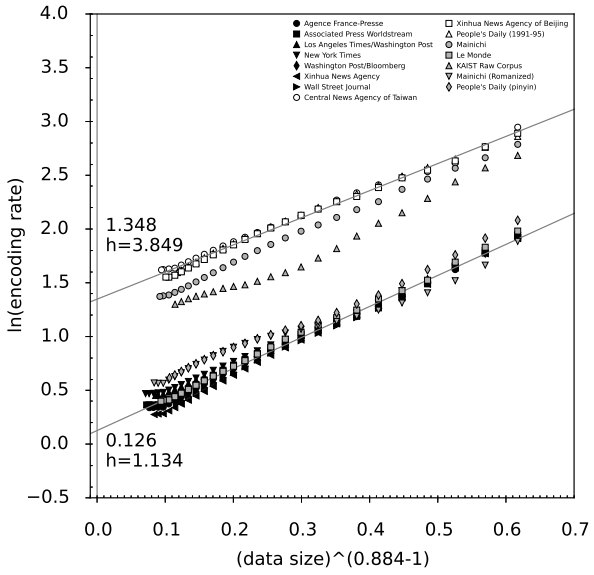

Figure 3: All large scale natural language data (first block of Table 1) from a linear perspective for function $f_3(n)$.

logarithm of the entropy rate. The intercepts are $h' = 0.126$ and $h' = 1.348$, with the corresponding entropy rates $h = 1.134$ bpc and $h = 3.849$ bpc, for the English and Chinese groups, respectively. Compared to the values reported previously, the entropy rate estimate h is smaller by 20%. Interestingly, a similar analysis can be conducted for ansatz $f_1(n)$. For this function, by using the average of $\beta = 0.789$, the final h was found to be 1.304 and 4.634 for English and Chinese, respectively, which is similar to previous reports. Therefore, the estimate of the entropy rate depends on the used ansatz, with the better fitting ansatz yielding estimates smaller than generally agreed.

Given our results, we may revisit the question whether the entropy rate of natural language is a strictly positive constant. Our estimates of the entropy were obtained through extrapolation. Thus, the possibility of a zero entropy rate cannot be completely excluded but it seems highly unlikely in view of the following remark. Namely, if the entropy rate is zero, then the data points should head towards negative infinity in Figure 3. However, the plots do not show such a rapid decrease for data size of the order of several gigabytes. On the contrary, all endings of the plots for large data sizes are slightly bent upwards. Hence we are inclined to believe that the true entropy rate of natural language is positive and close to our estimates. Of course, a far larger amount of data would be required to witness the behavior of the plots in the margin between the infinite limit and the largest data size considered in our experiment.

7 Conclusion

In this article, we have evaluated the entropy rates of several human languages by means of a state-of-the-art compression method. Compared to previous works, our contribution can be summarized as follows. First, we have calculated the compression rates for six different languages by using state-of-the-art corpora with sizes of up to 7.8 gigabytes. Second, we have extrapolated the empirical compression rates to some estimates of the entropy rate using a novel ansatz, which takes form of a stretched exponential function. This new ansatz function fits better than the previously proposed ansatzes and predicts smaller entropy rates than reported before. Especially for English, where the vast majority of previous works suggested an entropy rate around 1.3 bpc, our new results suggest the possibility of a value around 1.1 bpc. Some future extension of our work might be to simply enlarge the data, but it will not be trivial to obtain a uniform corpus of a larger scale. Hence, in the future work, it may be advisable to look for other computational approaches to the problem of entropy estimation.
The complete version of this article is available at (Takahira et al., 2016)

Acknowledgements

We like to thank Japan Science and Technology Agency (JST, Precursory Research for Embryonic Science and Technology) for financial support.

References

K. Atteson. 1999. The asymptotic redundancy of Bayes rules for Markov chains. *IEEE Transactions on Information Theory*, 45:2104–2109.

R. H. Baayen. 2001. *Word Frequency Distributions*. Kluwer Academic Publishers.

A. Barron, J. Rissanen, and B. Yu. 1998. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44:2743–2760.

T. C. Bell, J. G. Cleary, and I. H. Witten. 1990. *Text Compression*. Prentice Hall.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. C. Lai, and R. L. Mercer. 1983. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

T. M. Cover and R. C. King. 1978. A convergent gambling estimate of the entropy of English. *IEEE Transactions on Information Theory*, 24:413–421.

T. M. Cover and J. A. Thomas. 2006. *Elements of Information Theory*. Wiley-Interscience.

J. P. Crutchfield and D. P. Feldman. 2003. Regularities unseen, randomness observed: The entropy convergence hierarchy. *Chaos*, 15:25–54.

W. Ebeling and G. Nicolis. 1991. Entropy of symbolic sequences: The role of correlations. *Europhysics Letters*, 14(3):191–196.

D. Genzel and E. Charniak. 2002. Entropy rate constancy in text. In *Annual Meeting of the Association for the ACL*, pages 199–206.

P. Grassberger. 2002. Data Compression and Entropy Estimates by Non-sequential Recursive Pair Substitution. *ArXiv Physics e-prints*, July.

W. Hilberg. 1990. Der bekannte Grenzwert der redundanzfreien Information in Texten — eine Fehlinterpretation der Shannonschen Experimente? *Frequenz*, 44:243–248.

John C. Kieffer and Enhui Yang. 2000. Grammar-based codes: A new class of universal lossless source codes. *IEEE Transactions on Information Theory*, 46:737–754.

R. Levy and T. F. Jaeger. 2007. Speakers optimize information density through information density through syntactic reduction. In *Annual Conference on Neural Information Processing Systems*.

G. Louchard and W. Szpankowski. 1997. On the average redundancy rate of the Lempel-Ziv code. *IEEE Transactions on Information Theory*, 43:2–8.

C. G. Nevill-Manning and I. H. Witten. 1997. Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research*, 7:67–82.

B. Ryabko. 2010. Applications of universal source coding to statistical analysis of time series. In Isaac Woungang, Sudip Misra, and Subhas Chandra Misra, editors, *Selected Topics in Information and Coding Theory*, Series on Coding and Cryptology. World Scientific Publishing.

T. Schümann and P. Grassberger. 1996. Entropy estimation of symbol sequences. *Chaos*, 6(3):414–427.

S. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 30:379–423,623–656.

C. Shannon. 1951. Prediction and entropy of printed English. *Bell System Technical Journal*, 30:50–64.

Ryosuke Takahira, Kumiko Tanaka-Ishii, and Łukasz Dębowski. 2016. Entropy rate estimates for natural language—a new extrapolation of compressed large-scale corpora. *Entropy*, 18(10):364, Oct.

J. Ziv and A. Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343.