Study on Constants of Natural Language Texts

Daisuke Kimura† and Kumiko Tanaka-Ishii†

This paper considers different measures that might become constants for any length of a given natural language text. Such measures indicate a potential for studying the complexity of natural language but have previously only been studied using relatively small English texts. In this study, we consider measures for texts in languages other than English, and for large-scale texts. Among the candidate measures, we consider Yule’s $K$, Orlov’s $Z$, and Golcher’s $VM$, each of whose convergence has been previously argued empirically. Furthermore, we introduce entropy $H$, and a measure, $r$, related to the scale-free property of language. Our experiments show that both $K$ and $VM$ are convergent for texts in various languages, whereas the other measures are not.

Key Words: textual constants, multilingual texts, complexity, language models, redundancy

1 Introduction

Given documents or corpora collected under certain conditions, this article examines measures that potentially take invariant values for any size of text; we call such a measure textual constants. Textual constants were originally studied to provide a method for author identification. The oldest representative, to the best of our knowledge, is Yule’s $K$, proposed in 1940. Today, state-of-the-art techniques using language models or machine learning methods are more suitable tools for author identification. Textual constants, however, remain an interesting topic as they try to capture the characteristic of a text or corpus in an invariant value.

A text characteristic can represent the genre or difficulty of a text; natural language processing has formulated detection techniques for such characteristics. In the case of textual constants, as an original study targeting the author identification problem, researchers have devised different measures to quantify the complexity of vocabulary in terms of its richness and bias. In general, the larger the text, the more complex it is, but at the same time, if we consider a literary masterpiece such as Botchan by Soseki Natsume, any part should possess a characteristic identical to the whole. The prospect of representing this characteristic in terms of a value leads us to consider the consistent complexity underlying a text when treated as a word sequence. Moreover, the

† Graduate School of Information Science and Technology, University of Tokyo
* This article has been partially revised for better understanding of overseas readers.
document targeted here is not merely a single text but a set of texts of specific content. Would such a constant value for a set of texts not suggest the underlying properties of natural language?

As reported here, devising a statistic that should be constant is not an easy problem. One reason for this lies in the large proportion of the natural language text of hapax legomena—words occurring only once. For example, estimating probabilities for rolling dice is straightforward. In the case of a document, however, (Baayen 2001) showed that a textual constant must always be approximated with an insufficient estimate of word occurrences. In other words, developing a textual constant indicates considering a measure without either a solid language model or a sufficient amount of corpus data.

As we summarize in the next section, there have already been various proposals for textual constants, categorized as statistics based on words or on strings. According to a recent report, most such measures vary monotonically, with only two measures converging (Tweedie and Baayen 1998). In this context, the contribution of this study can be summarized in terms of the following four points. First, we show that one of the two measures suggested as a constant is, in fact not a constant. Second, in addition to examining existing measures, we propose new measures that attempt to capture the global properties of language in terms of complex networks and language entropy, and we demonstrate that these do not converge. In this sense, this study does not actually propose any new textual constants, and the measures we consider to be constants remain among those proposed so far. Third, with the exception of the previous work of (Golcher 2007), which considered various Indo-European languages, studies related to textual constants have only considered English. Hence, this study verifies textual constants for Japanese and Chinese texts as well. Fourth, previous studies have only considered fairly short texts for verifying constancy. In contrast, this article provides experimental results using texts as large as several hundred megabytes.

2 Related Work

There are two methods to define the textual constants in the previous studies, namely, word based and character-based approaches.

As mentioned previously, (Yule 1944) first proposed a textual constant based on word unigrams. Yule’s objective lay in author identification, for which he proposed $K$. With this background, (Herdan 1964) proposed his version of a textual constant, again for author identification. After various individual proposals, (Tweedie and Baayen 1998) investigated several word unigrams based methods. They considered 12 previously proposed measures and examined whether
they truly exhibit constancy using short English texts such as *Alice’s Adventures in Wonderland*. Since these 12 measures all hypothesize that words occur randomly, they did not experimentally analyze the texts as is, respecting word order, but instead shuffled word occurrences randomly before obtaining the values for each measure. They concluded that among the 12 measures, only $K$ and $Z$ become constant independent of the document length. Moreover, Tweedie and Baayen studied whether such constant values are usable for author identification. They plotted each document in $K$–$Z$ space and compared the classification possibility with other cluster analysis methods. They concluded that they could characterize a text using $K$ and $Z$ measures.

In this article, we reach a different conclusion from for measures based on word unigrams. Between $K$ and $Z$, we show that $Z$ is not a textual constant. As explained in the next section, $Z$ has a strong relation with complex networks. Under this view, we consider a simple measure $r$ as another possible constant but show that it is not a textual constant either. Although Tweedie and Baayen only considered short English texts, we verify constancy with Japanese and Chinese texts as well.

The entropy of language cannot be ignored in the context of textual constants. Since (Shannon 1948) proposed information-theoretic entropy, researchers have developed many novel ways to calculate language entropy, including string- and n-gram-based methods (Cover and Thomas 2006). Language entropy characterizes the redundancy of strings and can thus be expected to converge to an upper bound value. In the language processing domain, (Brown, Pietra, Pietra, Lai, and Mercer 1983) proposed methods to calculate the upper bound of language entropy but did not discuss how the calculated value would shift according to the data size. Such methods are difficult to adopt in our context, as they require estimating parameters using a subset of data. In another study, (Genzel and Charniak 2002) hypothesized that the entropy rate is constant. It is unknown, however, whether this entropy rate truly exhibits constancy, as their article suggests this possibility only by showing how the values of two mathematical terms forming the entropy rate both increase. Given this, to calculate the entropy value of text, we choose Farach’s method in (Farach, Noordewier, Savari, Shepp, Wyner, and Ziv 1995) to calculate the entropy value of test, as it does not require parameter estimation and will consider the constancy of language entropy by applying this method.

Finally, (Golcher 2007) recently showed how his value $VM$, which measures the repetition underlying a text, could be a textual constant. Although we discuss his approach in detail later, briefly, he showed how the ratio of the number of internal nodes of a text’s suffix tree to the total length of the text converges for texts in 20 different Indo-European languages. Moreover, he showed that the convergent value for all these languages is generally around 0.5, which differs
from values obtained for programming language texts. At the same time, Golcher also showed how random texts exhibit oscillation, unlike natural language texts. For example, oscillation is clearly observed in Figure 1, which shows the relation of the log of the number of characters (horizontal axis) and the value of VM (vertical axis), reproduced according to Golcher’s report. Following Golcher’s experiment, we consider that his VM has the potential to become a textual constant. He did not present a theoretical grounding of why VM should be constant, and this remains as future work from the perspective of this article. Although Golcher showed results only for Indo-European languages, we show experimental results for Japanese and Chinese texts as well and examine the potential of VM to be a textual constant.

3 Measures

As mentioned previously, we consider three word based measures $K$, $Z$, $r$ and two string based measures $VM$ and $H$. This section explains each of them in detail.

3.1 Word Based Measures

Yule’s Measure $K$

$K$ was introduced by Yule in 1944 to indicate the vocabulary richness (Yule 1944) of a given text. Given a text, let $N$ be the total number of words, $V$ be the vocabulary size (number of different tokens), and $V(m, N)$ be the number of words occurring $m$ times. Then, $K$ is defined as follows:

$$K = C \left[ -\frac{1}{N} + \sum_{m=1}^{m=N} V(m, N) \left( \frac{m}{N} \right)^2 \right].$$  \hfill (1)
Here, $C$ is simply a constant enlarging the final value of $K$, set as $C = 10^4$ by Yule. For the generative model of text, Yule assumes an urn model, in which words occur randomly. Under this model, when $N$ is sufficiently large, the expected value of $K$ can be mathematically proven to converge to a constant (Baayen 2001).

We briefly explain why $K$ indicates vocabulary richness as follows. Consider randomly choosing a word from a text randomly. In equation (1), $\left(\frac{m}{N}\right)$ gives the probability that a word occurs $m$ times in a text. Hence, $\left(\frac{m}{N}\right)^2$ gives the probability that the word is selected twice consecutively. When the probability of the same word being selected successively is large, vocabulary richness is limited, whereas when the probability remains small, the vocabulary will be large.

Equation (1) shows how $K$ becomes large in the former case but small in the latter case. To sum up, Yule’s $K$ is a measure indicating the vocabulary richness of text, according to consecutive occurrences of words.

**Zipf’s Law-Based Measure $Z$**

It is known empirically that the vocabulary distribution of a text follows Zipf’s law (Zipf 1949), and $Z$ is a measure based on Zipf’s law. Let $N$ be the total number of words, $V_N$ be the vocabulary size, and $z$ be a variable indicating the ranking of a word when all words are sorted in descending order of the number of occurrences (i.e., the most frequently occurring word has $z = 1$). Let $f(z,N)$ represent the frequency of the word of rank $z$. Then, it is known that the following scale-free relation holds for $f(z,N)$ and $z$ empirically:

$$f(z,N) = \frac{C}{z^a}$$ (2)

Here $C$ is a normalization term defined so that the equation $\sum_z f(z,N) = N$ holds. Moreover, setting $a = 1, C = V_N$ in equation (2), we can deduce that the number of words occurring $m$ times (i.e., $V(m,N)$) can be written as follows:

$$V(m,N) = \frac{V_N}{m(m+1)}.$$ (3)

Orlov et al. extended Zipf’s law and showed that the expected value of the vocabulary size $V_N$ for a text of length $N$ can be mathematically described using a sole parameter $Z$, as follows (Orlov and Chitashvili 1983):

$$E[V_N] = \frac{Z}{\log(pZ)} \frac{N}{N-Z} \log\left(\frac{N}{Z}\right).$$ (4)

Here $p$ indicates the largest relative frequency of words and is assumed to take almost the same constant value independent of any text.
$Z$ is the number of words when equation (3) best fits a given text. Moreover, by fixing $N$ to a set value in equation (4), we observe that $E[V_N]$ increases with increasing $Z$. Therefore, $Z$ can be interpreted as indicating the vocabulary richness of a text.

Finally, we consider the calculation method for $Z$. Replacing the expected vocabulary size $E[V_N]$ with the actual value $V_N$ for a text of length $N$ in equation (4) gives the following equation:

$$V_N = \frac{Z}{\log (pZ)} N - Z \log \left( \frac{N}{Z} \right).$$

(5)

This equation cannot be analytically solvable for $Z$. Therefore, to obtain $Z$, we set the following function $f(Z)$ to zero and use Newton’s method.

$$f(Z) = \frac{Z}{\log (pZ)} N - Z \log \left( \frac{N}{Z} \right) - V_N$$

(6)

Measure $r$ Based on Complex Networks

Given that $Z$ has a strong relationship with the scale-free property of texts, we introduce $r$, which measures that property more directly through a word network structure.

First, we explain the notion of $\Omega = (W, E)$, an undirected graph obtained for a text. Let $V$ be the vocabulary size; $W = \{w_i\}$ $(i = 1, \ldots, V)$ be the set of nodes in the graph where each node is a word; and $E = \{(w_i, w_j)\}$ be the set of edges, where an edge represents the successive occurrence of two nodes (i.e., words). In other words, the network considered here has nodes of words with branches indicating that two words occur successively.

In this article, we consider such a network constructed from words in a text. Apart from this approach, there are other possibilities for network construction based on word relations formed through phrase structure and co-occurrence. Since the focus here, however, is to consider the invariance underlying the scale-free property of language, these differences in approach should not drastically affect the conclusion. This was indeed the case for different network constructions that we considered, and therefore, we evaluate the word network mentioned above.

First, we consider the distribution of the degrees of nodes. Let the probability of a node of degree $k$ be $P(k)$. Figure 2 shows a log log graph of the distribution of degrees for a text written in English and programs written in Java. The horizontal axis indicates the log of the degree $k$, and the vertical axis indicates the log of $P(k)$. Since both plots form straight lines until a certain degree, both distributions follow a power law. This is the scale-free property of language and is observed in various complex network systems (Barabási and Albert 1999).

The power law distribution can be given as follows:

$$P(k) = ck^{-\gamma}.$$
Here $c$ is a normalization constant and can be obtained from the condition that $\sum_{k=1}^{\infty} P(k) = 1$. Taking the log of both sides of equation (7) gives

$$\log P(k) = -\gamma \log k + \log c,$$

thus showing the distribution’s linear appearance in a log log graph.

We then define the measure $r$ as the slope of this straight line given by formula (8), as follows:

$$r = -\gamma.$$  

There is no theoretical background for whether the value of this measure becomes constant or not. As mentioned for $Z$ in the previous section, however, since $r$ represents a power law property governing the global behavior of language, there is the possibility that the value becomes invariant, independent of the text length.

Finally, the calculation of $r$ in this study is conducted as follows. First, the word network is constructed from a text, and its degree distribution is obtained, as shown in Figure 2. Then, $r$ is obtained by minimizing the square error for the plots of degree from 2 to the smallest degree in the range of $\sum_{k=1}^{n} P(k) \geq A$.\footnote{A was set to 0.95 in our experiment.} This is conducted to ensure that the distribution does not follow a power law when the degree is either 1 or very large.

### 3.2 String Based Measures

**Golcher’s Measure $VM$**

(Golcher 2007) proposed $VM$ as a measure representing the repetitiveness of strings in a given text and calculated it using a suffix tree. This is a Patricia tree structure of a given string.
representing its suffixes. Let $S$ be a string, $T$ be its length, $S[i]$ be the $i$th character of $S$, and $S[i, j]$ be the substring from the $i$th to the $j$th characters ($i, j \in \{1, \ldots, T\}, i \leq j$). The suffix tree $T$ of string $S$ is then defined as follows (Ukkonen 1995).

When the directed tree $T$ from the root to the leaf meets the following conditions, $T$ is the suffix tree of a given string $S$;
- There are $T$ leaf nodes, labeled with integers from 1 to $T$.
- Each internal node has at least two children, with each branch labeled by a non-empty string included in $S$.
- The labels starting from a node always start with different characters.
- For every leaf $i$, the label from the root to the leaf is $S[i, T]$.

Golcher constructed a suffix tree by attaching a special character at the end of a string. For example, Figure 3 shows a suffix tree for the string “cocoa.”

$VM$ is defined using this suffix tree. Let $T$ be the length of string $S$ and $k$ be the number of internal nodes of the suffix tree constructed from $S$. Then, we define measure $VM$ as follows:

$$VM = \frac{k}{T}. \quad (10)$$

Since a suffix tree of length $m$ has $m$ leaves, the number of internal nodes is at most $m - 2$. Therefore, since $0 \leq k < T - 2$, the range of values for $VM$ is $0 \leq VM < 1$. According to Ukkonen’s algorithm, the number of internal nodes will increase with increasing number of repeated substrings. Therefore, $VM$ can be considered to represent the degree of repetitiveness underlying a given string.

Finally, we calculate $VM$ as follows in this article. The value of $VM$ defined in equation (10) requires obtaining the number of internal nodes of the suffix tree. The most straightforward method would be to construct the suffix tree directly and count the internal nodes. The required memory to construct a suffix tree is known to be many times greater than that for the original string, which is unrealistic since we consider large-scale data here. In this study, we instead use a

---

**Fig. 3** Suffix tree for the string “cocoa”
suffix array and obtain the number of internal nodes of the corresponding suffix tree by traversing
the array. The algorithm is detailed in (Kasai, Lee, Arimura, Arikawa, and Park 2001).

Text Entropy $H$

The information-theoretic entropy $H$ was introduced by (Shannon 1948). Let $\chi$ be the finite
set of alphabets, and $X$ be the random variable of $\chi$. Then, letting $P_X(x) = \text{Pr}(X = x)$ be the
probability of an alphabet $x \in \chi$, the entropy $H$ is defined as follows:

$$H = - \sum_{x \in \chi} P_X(x) \log P_X(x).$$

(11)

Direct calculation of equation (11) requires the probability $P_X(x)$ for each alphabet $x$. The
probabilities estimated from a text are approximations whose true values are unknown. In the
language processing domain, there have been various attempts to calculate the entropy of lan-
guage through texts such as those reported in (Cover and Thomas 2006; Brown et al. 1983).
In this study, we use Farach’s method, since it is theoretically proven to converge and does not
require any parameter estimation (Farach et al. 1995).

For this calculation, let $S$ be a string, $T$ be its length, $S[i]$ be the $i$th character of $S$, and
$S[i, j]$ be the substring from the $i$th to $j$th characters ($i, j \in \{1, \ldots, T\}, i \leq j$). For every position
$i$ ($1 \leq i \leq T$) in $S$, the largest matching $L_i$ is defined as follows:

$$L_i = \max\{k : S[j, j + k] = S[i, i + k] \mid j \in \{1, \ldots, i - 1\}, 1 \leq j \leq j + k \leq i - 1\}.$$  

(12)

In other words, $L_i$ is the longest common substring found between $S[1, i - 1]$ and $S[i, T]$. Let $\bar{L}$
be the average value of $L_i$:

$$\bar{L} = \frac{1}{T} \sum_{i=1}^{i=T} L_i.$$  

(13)

Then, the estimated entropy $H$ by Farach’s method is given as follows:

$$H = \frac{\log_{2}T}{\bar{L}}.$$  

(14)

If the true entropy value is $H_t$, this method is mathematically proven to give $|H_t - H| = O(1)$
when $T \rightarrow \infty$. 
4 Experiment

In this study, we verify whether the measures explained in Section 3 converge for data ranging in size from scores to almost 200 MB, and consisting of natural language texts and program texts written in a programming language.

Section 4.1 explains our data and the experimental setting. Then, Sections 4.2 and 4.3 provide the results for small- and large-scale data, respectively.

4.1 Data and Experimental Setting

4.1.1 Data

Table 1 lists the corpora used in this experiment. For the small scale corpora, unlike in a previous study (Tweedie and Baayen 1998), we included Japanese, French, and Spanish texts, in addition to English texts. The small scale data sources are listed in the first block of the table.

The previous study mentioned above (Tweedie and Baayen 1998) suggests that small-scale corpora are not sufficiently long to verify whether a measure converges. Therefore, we also used newspaper corpora in Japanese, English, and Chinese. Moreover, to verify the difference in the invariant values between natural and programming languages we compared the natural language results with those obtained using texts of the programming languages Java, Ruby, and Lisp.

In the cases of Japanese and Chinese, values were calculated for both the original texts

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Corpora used in this experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Small-scale data</strong></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>Title</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Japanese</td>
<td>Douhyou</td>
</tr>
<tr>
<td>English</td>
<td>Two Years Before the Mast</td>
</tr>
<tr>
<td>French</td>
<td>Robert Burns Vol. I</td>
</tr>
<tr>
<td>Spanish</td>
<td>El Criterio</td>
</tr>
<tr>
<td><strong>Large-scale Data</strong></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>Content</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Japanese</td>
<td>Asahi newspaper 2001</td>
</tr>
<tr>
<td>Japanese (roma-ji)</td>
<td>Roman transcription of Asahi</td>
</tr>
<tr>
<td>English</td>
<td>WSJ Corpus</td>
</tr>
<tr>
<td>Chinese</td>
<td>Beijing University Corpus</td>
</tr>
<tr>
<td>Chinese (pinyin)</td>
<td>Beijing University Corpus</td>
</tr>
<tr>
<td>Java</td>
<td>Source of Java packages</td>
</tr>
<tr>
<td>Ruby</td>
<td>Sources collected from Ruby Gems</td>
</tr>
<tr>
<td>Lisp</td>
<td>Various library sources of Lisp</td>
</tr>
</tbody>
</table>
and their romanized transcription, for measures $VM$ and $H$ only. For the other languages, all measures were calculated for the data listed in Table 1. As for the programming language texts, the program sources were separated into identifiers and operators, and every resulting unit was considered as a word. For example, the text “if(i < 5) break;” is considered to be a sequence of length 8; therefore, each unit, namely, “if,” “(,” “i,” “<,” “5,” “),” “break,” and “;,” is considered to be a word. In addition, parentheses were eliminated in the case of Lisp.

### 4.1.2 Applications Used for Preprocessing Data

In the experiment, we used certain public applications to preprocess the data, as discussed here. The word based measures $K$, $Z$, and $r$ require all texts to be transformed into word sequences. Thus, for word segmentation we used Mecab\textsuperscript{2} for Japanese and ICTCLAS\textsuperscript{3} for Chinese. For the string based measures $H$ and $VM$, we considered Japanese and Chinese in both the original texts and their romanized transcriptions, as mentioned before. For romanization, we used the pinyin transcription included with the data for Chinese, whereas for Japanese the texts were transcripted using KAKASI\textsuperscript{4}.

### 4.2 Results for Small Scale Corpora

The results for the small scale data are shown in Figures 4–8. In each figure, the horizontal axis indicates the log of the data size (in terms of the number of words or characters, depending on the measure), whereas the vertical axis indicates the measure values for each data source.

\begin{figure}[h]
\centering
\begin{subfigure}{0.5\textwidth}
\centering
\includegraphics[width=\textwidth]{fig4}
\caption{$K$ for small-scale corpora}
\end{subfigure}\hfill
\begin{subfigure}{0.5\textwidth}
\centering
\includegraphics[width=\textwidth]{fig5}
\caption{$VM$ for small-scale corpora}
\end{subfigure}
\end{figure}

\textsuperscript{2} http://mecab.sourceforge.net/
\textsuperscript{3} http://ictclas.org/
\textsuperscript{4} http://kakasi.namazu.org/index.html.ja
The results suggest that for all languages, including English, other Indo-European languages, and Japanese, $K$ and $VM$ converge, whereas the values of $Z$, $r$, and $H$ monotonically increase or decrease. In contrast to Tweedie and Baayen’s result, in which $Z$ converged (Tweedie and Baayen 1998), our experiment suggested that $Z$ does not consistently converge. Similarly, the related measure $r$ did not converge.

### 4.3 Results for Large Scale Corpora

(Tweedie and Baayen 1998) shuffled their texts before calculating the values of the various measures. This was conducted to ensure that the texts followed a mathematical assumption made for these measures. Here shuffling means randomizing the word order. The final value of a measure for a text was obtained as the mean value of 20 trials with shuffling and evaluation. Although it is questionable whether this shuffling would be necessary to observe the invariance of each measure, it is true that texts have local variances. Therefore, to compare how the measure
of the original text changes by shuffling, and also for comparison with the previous report of
(Tweedie and Baayen 1998), we present both shuffled and original text results for measures K
and Z.

Note that for each of the figures discussed here, the horizontal axis indicates the log of the
text length, whereas the vertical axis indicates the value for each measure.

Figures 9 and 10 show K for various corpora and the corresponding shuffled results, respec-
tively. For the natural language texts, K became constant in all cases. Although the programming
language text results fluctuated slightly compared with the natural language results, the values
became almost stable after 100,000 words. Furthermore, the values for the shuffled results con-
verged for any language, although the value of K did not vary between data with and without
shuffling. Since K assumes the random occurrence of words but the randomness underlying natu-
ral language text is not evident, it is interesting that K was almost the same for the original text
as for the shuffled version. Note also that the values of K for the programming languages became
far larger than the values for the natural languages, demonstrating that K clearly distinguishes
between natural and programming languages.

For VM, we first show the results for English, and Japanese and Chinese in romanized tran-
scriptions. Figure 11 shows VM for each of the corpora, including the programming language
texts and Figure 12 is an enlarged version of Figure 11. The value for Japanese was slightly
larger than those for English and Chinese, but it almost converged to a value of 0.5. For the
programming languages, the values fluctuated more than for the natural languages; however,
these values did not show a monotonic trend. The final value of VM was larger than that for the
natural languages, at almost 0.65. This difference between natural and programming languages

![Fig. 9 K for large-scale corpora](image1)

![Fig. 10 K for large-scale corpora (mean after shuffling)](image2)
shows that the repetitiveness underlying natural languages is smaller than that underlying programming languages.

Next, the VM results for Japanese and Chinese in their original writing systems are shown in Figure 13, with an enlarged version in Figure 14. These figures also include the romanized transcription results for each language.

The values of VM showed a convergence tendency, even with the original writing systems in both Japanese and Chinese. The actual convergent values were approximately 0.35, which is smaller than those for the romanized cases. The reason for this is the far larger alphabet size in each of the original writing systems, which decreases the number of repetitive sequences found within a text and hence the number of internal nodes in the suffix tree as well.

For the results for the other three measures, Z, r, and H, Figures 15, 17, and 18 show
the respective results. In addition, Figure 16 shows the means of the randomly shuffled results corresponding to Figure 15. The measures related to complex networks, $Z$ and $r$, increased monotonically with the text length, with one exception: $Z$ for the programming language text in Lisp, which increased only slightly and almost converged.

$Z$ and $r$ captured the global structure underlying language, but their results in general did not converge to a value. Moreover, the entropy $H$ monotonically decreased. Note that the results here for Japanese and Chinese are for romanized transcriptions. Furthermore, with the exception for Lisp noted above, the results for $Z$ did not converge even after the shuffling the texts, as shown in Figure 15. It is interesting that $Z$ only showed the different tendency of a slight increase in the case of Lisp.

In addition to the experimental approaches described above, we obtained results for $VM$ and $H$ when the texts were reversed, as these measures depend on the order of a text. We obtained
the same results for both normal and reversed orders.

5 Discussion

First, we consider the convergence of the measures. From the results reported in the previous section, only $K$ and $VM$ exhibited stable convergence.

In the case of $VM$, the value reached approximately 0.5 for all languages, including the romanized transcriptions for Japanese and Chinese (but not the original texts). This value might signify the complexity underlying natural language when considered via an alphabet as a rough approximation of phonetic sounds. The values of $VM$ were smaller for Japanese and Chinese with the original writing systems, because of the decrease in repetition. In other words, when a writing system has a larger number of units, the redundancy underlying language becomes smaller. Moreover, it is interesting to observe that a measure that oscillates for random sequences (as shown in Section 2) becomes stable for natural language.

(Tweedie and Baayen 1998) showed convergence of $Z$ for short English texts, but in our results, the value increased monotonically and did not converge for either small scale or large scale texts, except for the Lisp program corpus that showed only a slight increase. As in this case, some measures show convergence depending on the corpus. Since this work seeks a textual constant demonstrating universal convergence independent of different language corpora, $Z$ does not qualify as a textual constant in this context. It would be interesting, however, to verify which measures converge for which type of data, which remains as future work.

Regarding another measure related to complex networks, $r$, the values shifted with the text lengths. In the case of the entropy $H$, the values did not converge even when using a calculation method with mathematical proof of convergence. Since $H$ can be calculated through n-grams, and $VM$ is also calculated through n-grams via a suffix array here, we suspected that $H$ and $VM$ might present a similar tendency. However, in this study, they yielded different results.

Finally, it is interesting to discuss the discriminatory nature of the measures. Description could be provided at different levels of texts, authors, genres, languages, language families, and natural/artificial languages. $K$ and $VM$ showed clear differences between the convergent values for natural languages and programming languages. In fact, such differences were apparent for all the measures, demonstrating an essential difference underlying the complexities of these two types of languages. Language families and languages could be distinguished through the alphabet size, at least for $VM$. (Tweedie and Baayen 1998) suggested that texts were discernable through both $K$ and $Z$, but concluded that state-of-the-art machine learning techniques are more suitable
for solving such problems. To summarize, then, the computational linguistic signification of the study of textual constants in a text can be concluded to represent the degree of redundancy underlying natural language.

6 Conclusion

In this study, three previously proposed text constancy measures \((K, Z, VM)\) and two new measures \((r, H)\) were studied for convergence for both natural and programming languages when the amount of data was increased. Yule’s \(K\) is a classic measure representing the richness of vocabulary, whereas Orlov’s \(Z\) and \(r\) are related to complex networks. \(VM\) represents the repetitiveness underlying text, as measured through a suffix array, and \(H\) is the entropy of a text. This article introduces \(r\) and \(H\) for the first time in this context.

In our experiment, the measures were extensively verified for a number of small and large scale texts from both natural and programming language sources. The results suggested that \(K\) and \(VM\) were almost convergent. Moreover, these two measures exhibited significant differences in the convergent values, whereas, the other three measures varied with text length.

Compared with the previous study of (Tweedie and Baayen 1998), which reported that both \(K\) and \(Z\) are convergent, we observed convergence for \(K\), but not for \(Z\). Moreover, compared with (Golcher 2007), we observed convergence for romanized transcriptions of Japanese and Chinese to 0.5, almost the same value as those for Indo-European languages.

Acknowledgement

We thank Prof. Fumiyasu Komaki from the Graduate School of Information Technology Science, the University of Tokyo, for his valuable comments on statistical tests.

Reference


**Daisuke Kimura:** Daisuke Kimura is a Ph.D student at the Graduate School of Information Science and Technology, the University of Tokyo. He graduated from the Department of Mathematical Engineering and Information Physics, School of Engineering, the University of Tokyo, in 2010 and completed a masters program from the above university in 2012. His interest lies in machine learning and succinct data structure.

**Kumiko Tanaka-Ishii:** Kumiko Tanaka-Ishii is a professor at the Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University. After receiving her Ph.D. from the University of Tokyo, she was a
faculty member of the University of Tokyo, and moved to Kyushu University in 2012. Her main interest lies in computational linguistics and natural language processing.

(Received August 5, 2010)
(Revised December 20, 2010)
(Accepted February 10, 2011)