Zipf's Law and Heaps' Law Can Predict the Size of Potential Words

Yukie SANO,[1] Hideki TAKAYASU[2,3] and Misako TAKAYASU[4]

[1]*College of Science and Technology, Nihon University, Funabashi 274-8501, Japan*
[2]*Sony Computer Science Laboratories, 3-14-13 Higashi-Gotanda, Tokyo 141-0022, Japan*
[3]*Meiji Institute for Advanced Study of Mathematical Sciences, Kawasaki 214-8571, Japan*
[4]*Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama 226-8502, Japan*

We confirm Zipf's law and Heaps' law using various types of documents such as literary works, blogs, and computer programs. Independent of the document type, the exponents of Zipf' law are estimated to be approximately 1, whereas Heaps' exponents appear to be dependent on the observation size, and the estimated values are scattered around 0.5. By definition, randomly shuffled documents reproduce Zipf's law and Heaps' law. However, artificially generated documents using the empirically observed Zipf's law and number of distinct words do not reproduce Heaps' law. We demonstrate that Heaps' law holds for artificial documents in which a certain number of distinct words are added to empirically observed distinct words. This suggests that the number of potential distinct words considered in the creation of a given document can be predicted.

§1. Introduction

Zipf's law[1)] is an empirical law stating that word frequency in documents is inversely proportional to word rank in descending order of occurrence. Zipf's law is consistent with the following power law cumulative distribution of the number of word appearances:

$$P(\geq x) \propto x^{-\alpha}, \tag{1·1}$$

where the exponent α is a positive constant. Zipf's law is applicable not only to word frequencies in documents but also to incomes of firms and individuals, the sizes of gypsum fragments, and the abundances of expressed genes.[2)–4)] As such, Zipf's law has attracted a great deal of attention, but the general mechanism of Zipf's law remains unclear. In quantitative linguistics, Zipf's exponent, α, is evaluated for various languages, such as English and Russian,[5)] and for programming languages, such as Java and C,[6)] and is known in many cases to be approximately 1.

Heaps' law[7)] states that the number of distinct words increases nonlinearly as the total number of words in a document increases. The number of distinct words $D(n)$ among the first n words of a document is approximated by the following power law:

$$D(n) \propto n^{\beta}. \tag{1·2}$$

Heaps' exponent, β, is smaller than 1, and Araújo et al. estimating from newspaper articles in the Wall Street Journal and from scientific papers concluded β is 0.4

to 0.6.[9] Based on several thousands of web pages, Baldi et al. estimated that $\beta = 0.76$,[10] and Zhang reported that, in computer programming tokens, which include several identifiers, such as ":", β takes a value of between 0.540 and 0.869.[6]

Baeza-Yates et al. reported that Zipf's law and Heaps' law are equivalent and that $\beta = \alpha$ when $\alpha < 1$.[11] Leijenhorst et al. developed Heaps' law from the Zipf-Mandelbrot law in a more sophisticated manner and obtained the same result.[13] Lü et al. showed by analytical and simple numerical simulation that, in the case of $\alpha \geq 1$, the value of β is equal to 1.[12] Cattuto et al. observed these two empirical laws in social bookmark data that focused on word tag co-occurrence distributions for Zipf's law and Heaps' law curve for number of distinct word tags. Zipf's law and Heaps' law yielded the same results, i.e., $\beta = \alpha = 0.7$, for both empirical data and simple network simulations.[14),15] Furthermore, when applied in ecology especially in island biogeography, Zipf's law is valid for species abundance distributions and Heaps' law describes species-area relationships in which the number of species found within an area increases nonlinearly for increasing area.[16)–18] In this case, $\beta \simeq \alpha \simeq 0.5$.[17] Note that, although the total number of species can be estimated based on the size of the area, species-area relationships are not fully equivalent linguistics with respect to Heaps' law. Heaps' law is also known as rarefaction curve in ecology.[19),20] In this case, one can plot the number of species as a function of individuals sampled. Although it does not assume Zipf's law, it allows the calculation of the species richness for a given number of sampled individuals. Consequently, it is an important issue to preserve biodiversity.

In this paper we describe our data and show empirical results from the data in §2. In §3 we check the validity of Heaps' law under the condition of words from shuffled real documents and artificially generated documents by comparing empirical results. Finally in §4 we indicate predictability of the size of potential words in a certain type of document from the simulation.

§2. Empirical results

In this section, we confirm the validity of Zipf's law and Heaps' law for various types of real documents. We chose and analyzed a range of documents, covering various languages, professional and amateur authors and human spoken natural languages and computer processing formal languages.

- "The Adventures of Sherlock Holmes" was written in English by Sir Arthur Conan Doyle during the 1890s and is one of most frequently downloaded documents from Project Gutenberg, which freely provides public domain book content on the web.[21]
- "Don Quixote" was written in Spanish by Miguel de Cervantes during the 16th century and is the most frequently downloaded book in the Spanish category of Project Gutenberg.
- "Light and Darkness (Meian)" was written in Japanese by Soseki Natsume during the 1910s. This unfinished novel is the longest work of Soseki Natsume, who is one of the most popular novelists in Japanese history. Although this novel is unfinished, we got similar results from his novel "Kokoro" and "I am a

Table I. Data description of documents.

	Language	Total words N	α	β	β_{rand}
Sherlock Holmes	English	107,219	0.97	0.54	0.52
Don Quixote	Spanish	390,402	0.98	0.65	0.57
Light and Darkness	Japanese	180,623	1.04	0.49	0.46
Blog	Japanese	308,687	0.98	0.57	0.50
NumPy	Python	850,699	1.10	0.68	0.50

Cat (Wagahai wa neko de aru)".

- Blog entries posted by a single anonymous blogger in Japanese who was randomly selected from among 20 thousand bloggers. We collected 5,000 blog entries posted by the author from 2007 to 2010 and confirmed that the blog was not a typical spam blog. We accumulated his 5,000 blog entries into one document in sequence of time.
- "NumPy", which is an open-source Python program developed by several contributors to provide support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays. NumPy includes 397 small programs.[22] We ignored flows of program procedures, such as loops and concatenated all Python programs in the library into one document. We also removed all comments written in natural language sentences but did not remove symbols.

The details of the documents are listed in Table I. In the case of Japanese, it is necessary to separate words, which are not separated by spaces. We use the "MeCab" morphological analyzer with a dictionary that added new words from Wikipedia titles and Hatena keywords' titles on March 2011 to accurately separate words, including new words from blogs. In all of the documents except NumPy, we removed all symbols, such as periods and colons, from the documents.

Figures 1 and 2 demonstrate the validity of Zipf's law and Heaps' law based on empirically obtained data and estimated exponents are shown in Table I using the Gauss-Newton algorithm to minimize the sum of the squares of the errors in the whole area under the graph. As expected, Zipf's exponent α is approximately 1 in all cases, which is confirmed to be universal. On the other hand, Heaps' exponent β takes values in the range of from 0 to 1. Although our results are estimated by linear scale in whole areas, the area of large n dominates and thus the Heaps' exponent tends to be close to 0.5. On the other hand, previous studies, for example,[5] are often estimated with an accuracy close to 1 by using log-log scale and the areas of small n are playing a central role to the estimation. Note that NumPy consists primarily of small programs developed by numerous different authors, the same names of variables tend to appear together in the same programs. Therefore, unlike for other spoken languages, the curve is not smooth and depends on the order of concatenation. In order to avoid this, we averaged 10 different patterns of Heaps' curves for concatenation.

Finally, we want to clarify the importance of having a limit of words on Heaps' law which states that distinct words grow by power functions in empirical data.

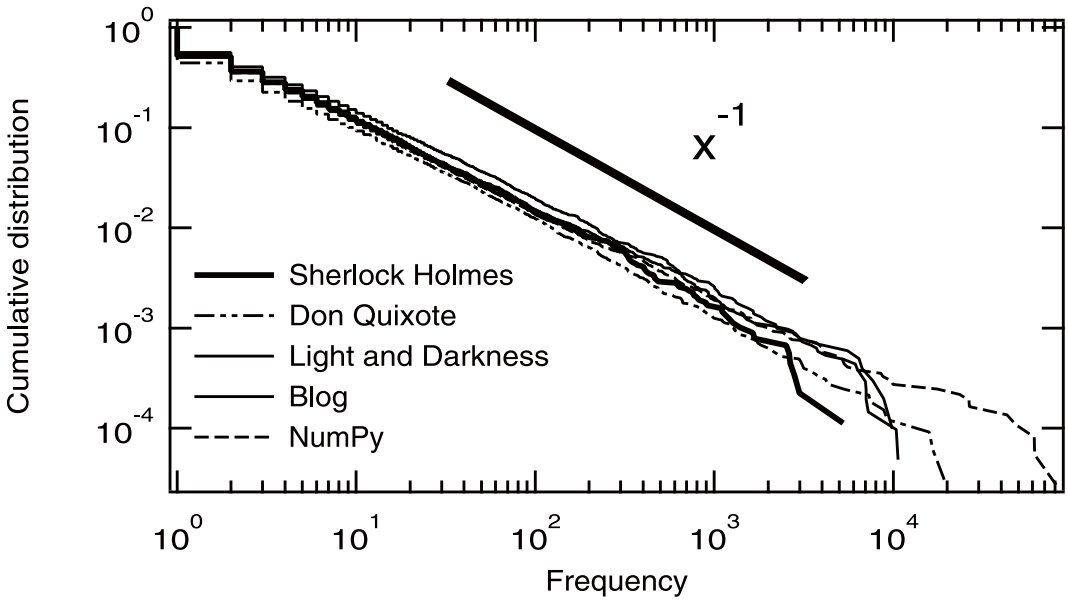

Fig. 1. Zipf's law for an English novel, a Spanish novel, a Japanese novel, a Japanese blog, and a computer program. Zipf's exponent α is approximately 1 in all cases.

While the number of words is limited in languages, there should be a point of saturation in very long documents. Therefore, when $n \to \infty$, $\beta \to 0$. On the other hand, at the beginning of the documents, new distinct words tend to appear one after another. Thus, $\beta \to 1$. In other words, Heaps' exponent depends on the value of n, and, for very long documents, a single Heaps' exponent cannot describe the entire document.

§3. Validation of Heaps' law

In this section, using the empirical data of the present study, we confirm that Heaps' law cannot be simply derived from Zipf's law.

3.1. *Shuffled documents*

As mentioned in previous studies,[10)–13)] if Heaps' law can be derived from Zipf's law alone, then we should be able to reproduce Heaps' law from randomly shuffled documents that ignore correlations between word occurrences. The circles in Fig. 3 indicate the shuffled results for the case of a blog which we shuffled 10 times with different random seeds and used the average. Since Heaps' exponents for a randomized document β_{rand} are almost the same as the value shown in Table I, Heaps' law can be regarded as the derivative of Zipf's law.

3.2. *Simulation using empirical Zipf's law*

Here, we generated artificial words using Zipf's law with the same empirically observed values of α and the same total number of words N and number of distinct

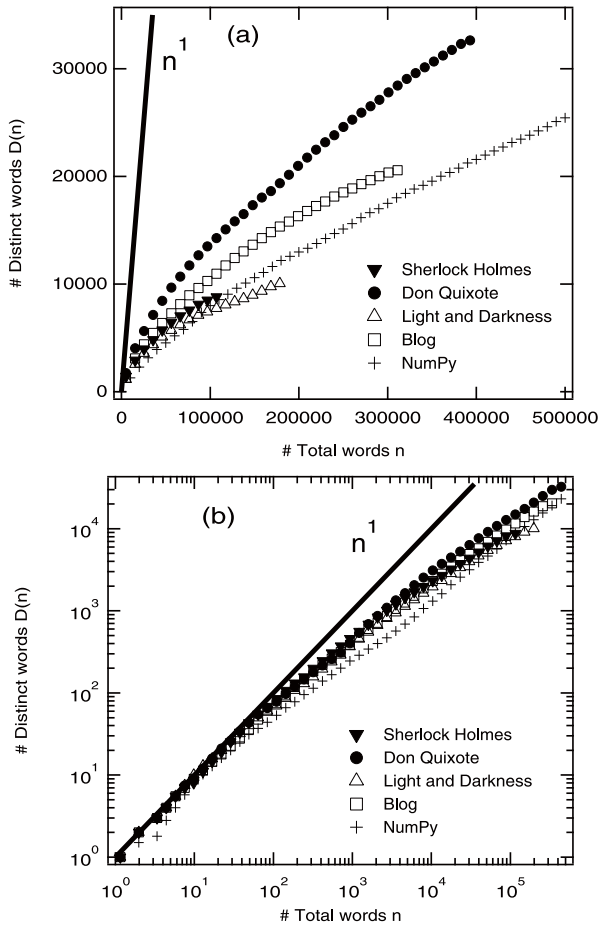

Fig. 2. Heaps' law for an English novel, a Spanish novel, a Japanese novel, a Japanese blog, and a computer program. Data is plotted (a) linearly and (b) on a log-log scale.

words $D(N)$. The author of the document is assumed to have $D(N)$ distinct words in his mind, and these words are assumed to be selected randomly with a probability based on the observed Zipf's law. The triangles in Fig. 3 (Simulation 1) show the results of this simulation using blog data. There is clearly a significant difference from the real data.

Table II. Number of observed distinct words $D(N)$ in documents containing a total of N words and $D^*(N)$ optimized distinct words in simulations.

	Distinct words $D(N)$	Simulated distinct words $D^*(N)$
Sherlock Holmes	8,910	14,800
Don Quixote	32,687	77,800
Light and Darkness	10,132	13,100
Blog	19,324	34,300
NumPy	36,676	101,900

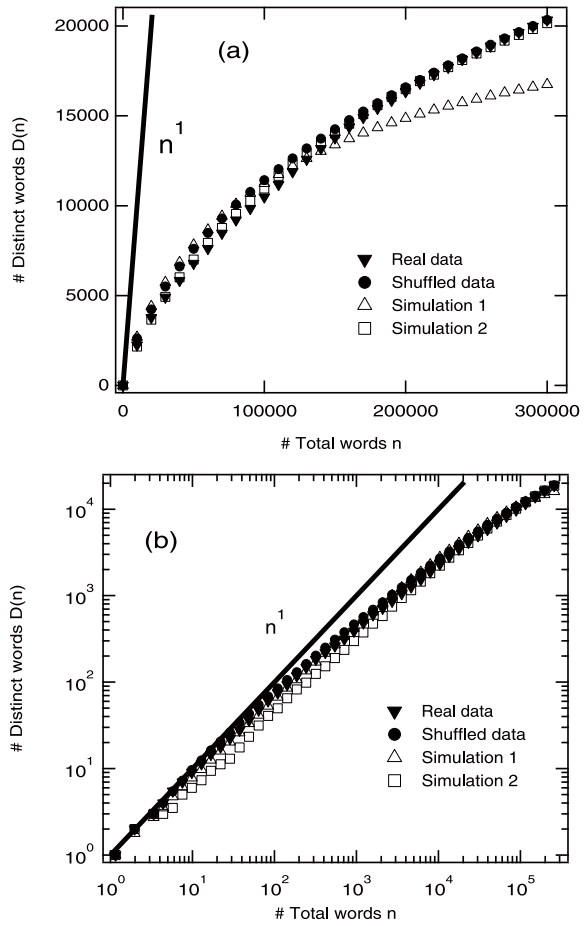

Fig. 3. Heaps' law for a Japanese blog (downward triangles), a randomly shuffled document (circles), simulation using a real number of distinct words $D(N)$ (triangles, Simulation 1), and simulation using a number of optimized distinct words $D^*(N)$ (squares, Simulation 2). Data is plotted (a) linearly and (b) on a log-log scale.

In order to clarify the reason for this discrepancy, we vary number of distinct words $D(N)$ while leaving Zipf's exponent α and the total number of words N unchanged. We select the number of distinct words $D^*(N)$ so as to minimize the sum of the squares of the residual errors. The squares in Fig. 3 (Simulation 2) indicate the modified simulation results, which are much closer to the results for the real data. The potential number of distinct words, $D^*(N)$, can be estimated in this manner for other examples, as summarized in Table II and Fig. 4. In any case, Heaps' law can be well reproduced for all examples by simulation using the optimized parameters, $D^*(N)$.

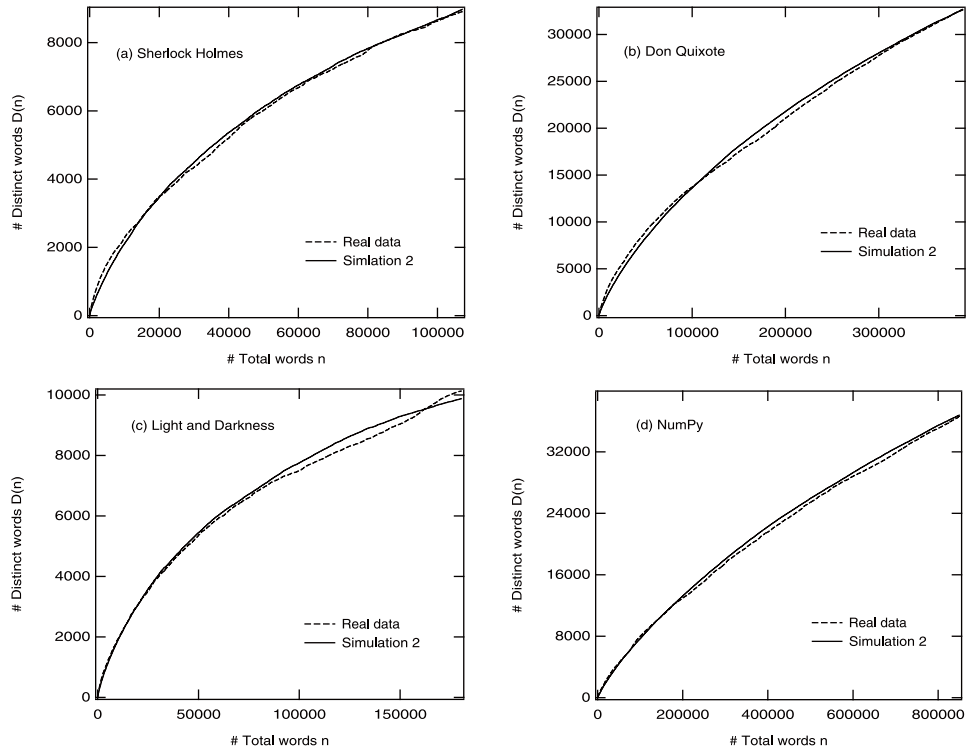

Fig. 4. Comparison of real data (dashed line) and simulated data obtained using the optimized parameters listed in Table II (solid line) for Heaps' law. (a) "The Adventures of Sherlock Holmes", (b) "Don Quixote", (c) "Light and Darkness (Meian)", and (d) NumPy.

§4. Summary and discussion

In the present study, we examined the validity of Zipf's law and Heaps' law for various types of documents. With respect to the shapes of the functions, Heaps' law is regarded as a deviation of Zipf's law, as reported in a previous study. We showed that a random sampling of N words from $D(N)$ distinct words cannot reproduce Heaps' law. However, Heaps' law can be revised using an estimated potential number of distinct words, $D^*(N)$. The obtained results indicate that we can estimate the potential number of words considered in the creation of given documents using Zipf's law and Heaps' law with real data.

Acknowledgements

The authors would like to thank Dentsu Kansai Inc. and Hottolink Inc. for providing the blog data. The present study was supported in part by a Grant-in-Aid for Scientific Research No. 22656025 (M.T.) from MEXT, Japan.

References

1) G. K. Zipf, *Human behavior and the principle of least effort* (Addison-Wesley, Cambridge, 1949).
2) K. Okuyama, M. Takayasu and H. Takayasu, Physica A **269** (1999), 125.
3) C. Furusawa and K. Kaneko, Phys. Rev. Lett. **90** (2003), 088102.
4) M. E. J. Newman, Contemporary Physics **46** (2005), 323.
5) A. Gelbukh and G. Sidorov, Lecture Notes in Computer Science **2004** (2001), 332.
6) H. Zhang, Information Processing and Management **45** (2009), 477.
7) H. S. Heaps, *Information Retrieval: Computational and Theoretical Aspects* (Academic Press, Orlando, 1978).
8) R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval* (ACM Press, 1999).
9) M. D. Araújo, G. Navarro and N. Ziviani, in *Proc. Fourth South American Workshop on String Processing*, Carleton University Press International Informatics Series **8** (1997), 2.
10) P. Baldi, P. Frasconi and P. Smyth, *Modeling the internet and the web: Probabilistic methods and algorithms* (Wiley, 2003).
11) R. Baeza-Yates and G. Navarro, J. of the American Society for Information Science **51** (2000), 69.
12) L. Lü, Z.-K. Zhang and T. Zhou, PLoS ONE **5** (2010), e14139.
13) D. C. van Leijenhorst and Th. P. van der Weide, Inf. Sci. **170** (2005), 263.
14) C. Cattuto, V. Loreto and L. Pietronero, Proc. Natl. Acad. Sci. **104** (2007), 1461.
15) C. Cattuto, A. Barrat, A. Baldassarri, G. Schehr and V. Loreto, Proc. Natl. Acad. Sci. **106** (2009), 10511.
16) O. Arrhenius, J. Ecol. **9** (1921), 95.
17) H. Irie, K. Tokita and H. Habara, Publ. RIMS, Kyoto Univ. **1432** (2005), 116 (in Japanese).
18) H. Irie and K. Tokita, q-bio/0609012.
19) R. Colwell and J. Coddington, Philos. Trans. R. Soc. London B **345** (1994), 101.
20) M. Arumugam, J. Raes, E. Pelletier, D. Le Paslier, T. Yamada, D. R. Mende, G. R. Fernandes et al., Nature **473** (2011), 174.
21) Digital libraries providing public domain books.
Gutenberg Project, `http://www.gutenberg.org/`
and Aozora bunko, `http://www.aozora.gr.jp/`
22) Extension to the Python programming language.
NumPy (Version1-6-1), `http://numpy.scipy.org/`
23) List of Japanese common new words.
Wikipedia titles, `http://ja.wikipedia.org`
and Hatena keywords, `http://d.hatena.ne.jp/keyword/`