The Computer Program for the Treatment of Big Data in the Field of Literature Science

Liliia Bodnar[1], Kateryna Shulakova[2] and Olena Tyurikova[3]
[1]*Department of Innovative Technologies and Methods of Teaching Natural Sciences, South Ukrainian National Pedagogical University, 26 Staroportofrankyvska Str., Odessa, Ukraine*
[2]*Department of Computer Engineering and Information Systems, State University of Intellectual Technologies and Telecommunications, 1 Kovalska Str., Odessa, Ukraine*
[3]*Department of Architectural Environment Design, Odessa State Academy of Civil Engineering and Architecture, 4 Didrihsona Str., Odessa, Ukraine*
bodnarl79@pdpu.edu.ua, k.shulakova@onat.edu.ua, tulenanik@mail.ru

Keywords: Program for Big Date Treatment, Zipf's Laws, Evolution of Languages.

Abstract: The problem of processing large databases is important for solving many pressing problems of science and technology. In this paper, we have developed a computer program (Conan 3.0) for processing large text arrays. However, other applications are possible. We have applied the developed program for the analysis of large texts on the basis of Zipf's laws. The task, which was solved in this work, is connected with the laws of the evolution of languages; in particular, correlations in the development of different Slavic languages were traced. It was assumed that an important characteristic of the language is the Zipf's constant. As a result of calculating the changes in the short-circuit over the 18th, 19th and 20th centuries for the Ukrainian, Russian and Polish languages, no significant changes in the short-circuit were revealed. Small fluctuations in the short-circuit for these languages do not correlate.

1 INTRODUCTION

A wide range of research [1, 2] has shown that tools from information theory (e.g. information content/surprises, entropy) are useful tools in addressing questions of linguistic interest. These range from predicting the targets and outcomes of phonological and syntactic processes, to explaining and evaluating models of linguistic data.

Zipf's law is a fundamental paradigm in the statistics of written and spoken natural language. Zipf's law usually refers to the fact $P(s) = Pr\{S > s\}$ that the value S of some stochastic variable, usually a size or frequency, is greater than s, decays with the growth of s as $P(s) \sim s^{-1}$. This in turn means that the probability density functions P (s) exhibits the power law dependence in (1):

$$P(s) \sim 1/s1+m \text{ with } m=1. \quad (1)$$

Zipf's law is strictly valid if randomly if a balanced condition is fulfilled: the sum of all the mechanisms responsible for the growth and decline of firms must vanish on average in a precise sense. Any departure from this requirement yields a departure of the tail index from its canonical value $m=1$. This result can allow one to understand why different tail indexes are reported in the literature for different countries around the world [3].

According to Zipf's law, in a list of word forms ordered by the frequency of occurrence, the frequency of the r th word form obeys a power function of r (the value r is called the rank of the word form). It should be noted that further surveys [4] showed that Zipf's law is roughly realized only for the most frequent words.

In Ref. [5] it was shown that evolution of languages connected with a biological capacity shared by all humans and distinguished by the central feature of discrete infinity – the capacity for unbounded composition of various linguistic objects into complex structures. These structures are generated by a recursive procedure that mediates the mapping between speech- or sign-based forms and meanings, including semantics of words and sentences and how they are situated and interpreted in discourse. This approach distinguishes the biological capacity for language from its many

possible functions, such as communication or internal thought.

The study of language evolution is performed using approaches of "Big Data". Different models of language evolution are expressed in multiple empirical domains. Databases for linguistic structure are available in the Internet [6].

In Ref. [7] it have explored the effectiveness of authorship attribution on works of literature. A certain authors have a highly recognizable style. It was considered usage statistics for the commonly used style markers for two authors. Each number is the number of function occurrences that is the particular function word. It was realize like our ways with usage Natural Language Toolkit (NTLK) library.

Margins, column widths, line spacing, and type styles are built-in. Some components, such as multi-levelled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

2 DESCRIPTION OF COMPUTER PROGRAM

The developed computer program (Conan 3.0) was created for analyzing large texts. There is also a possibility to use the program for assessing the quality of literary translations [9].

The program Conan 3.0 is based on the earlier version Conan 2.0, which was written in the C # language using the functionality of MSSQL database like Full-text Search for ordering and processing data which was getting after parsing of a text document and importing them into MSSQL Database. Such a way of processing data was pretty complicated and had a lot of restrictions and as a result of pretty low performance and quality of calculation [8] [9]. This led to the development of a new version of the program, where we abandoned the use of the database functionality in favor of dynamic analysis based on algorithms provided by the NTLK library.

Creating a new version of the application we were following such goals as:

- Make the application more native to the internet surrounding.
- Make the application usage easier for users.
- Increase performance and quality of calculation due to using Python and already well-developed python BigData libraries.

The program is based on the simple text data processing algorithm (Figure 1).

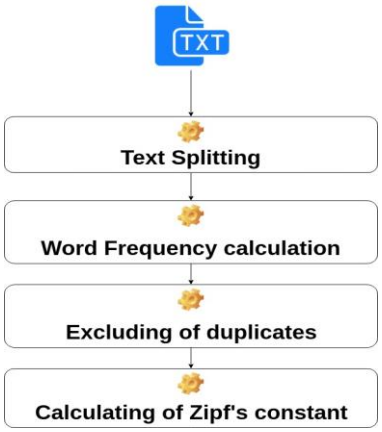

Figure 1: Algorithm for calculation of Zipf's constant.

Our previous programs did not take into account the important features of Zipf's laws. For example, the same word was accounted many times with different suffixes.

In the program Conan 3.0 the probabilistic search was introduced which made the calculations more accurate.

Based on the modern technologies for processing large amounts of data and the Natural Language Toolkit (NTLK) library [10], it was possible to rewrite the program code while using simple algorithms.

The data of this library were used also in the work [7].

NTLK includes extensive software, and documentation that we downloaded from source[1]. We used Natural Language Processing to provide any kind of computer manipulation with natural language. As to programming environment, the choice was done between two scripting languages, Python and JavaScript. At present much of server applications are written in JavaScript and Python languages [12]. Using each of the platforms we easily develop and maintain web applications of any complexity. Event-oriented architecture Node.js, which allows us to handle large streams of data simultaneously, and Python, in turn, is perfect for processing large amounts of data.

[1]http://www.nltk.org/

In this work the results were obtained using some approaches in the field of Big Data [9]. Thus, the chosen approach provided a platform for writing the program as a full-fledged web application for online use.

In the course of the chosen solution, it also became clear that there is no function that calculates the Zipf's constant in its pure form in this library. In the previous versions of the program, the main difficulty of data processing was not the calculation of the Zipf's constant, but the preparation and processing of the text under study. Therefore, the Natural Language Toolkit came up for the intermediate values needed for the Zipf's constant, such as calculating the frequency of occurrences of words. As a result, the algorithm for determining the Zipf's constant has not changed except for using the Natural Language Toolkit library instead of using the Full Text Search technology previously used in the full text search. This greatly accelerated the calculation of the Zipf's constant.

3 ANALYSIS OF SLAVIC LANGUAGES EVOLUTION FROM THE 18TH TO THE 20TH CENTURIES

The study was conducted in order to trace the evolution of the languages of the East Slavic and West Slavic groups: Ukrainian, Russian and Polish.

The representation in terms of the frequency distribution *f (n)* was successfully used to demonstrate the stability of the Zipf's constants for various literary works from the 18th to the 20th centuries with different sizes of texts.

Such works as “Litopis Samovidtsya” (18th century), “Compositions of Shevchenko” (19th century), “Compositions of Dovzhenko” (20th century) and others were investigated in Ukrainian.

Processing the text data for the Ukrainian language, the following results were obtained, shown in Figure 2.

The obtained data indicate that the Zipf's constant for Ukrainian works is within:

- 18th century from 0.049 to 0.072
- 19th century from 0.056 to 0.087
- 20th century from 0.05 to 0.061

The largest variation occurred in the 19th century at 0.031 units. Taking into account the fact that the Zipf's constant is a constant for each language, it can be assumed that this language was intensively modified in a given period of time, which led to a change in the identity of the language. Taking the fact that the value in the 20th century almost returned to the framework of the 18th century and the spread became much smaller, we can assume that destructive changes in the language did not occur and its identity was preserved in one way or another.

Studies of Russian works have shown the results, which are presented in Figure 3. Such works as "Works of Lomonomov" (18th century), "Works of Tolstoy" (19th century), "Works of Bunin" (20th century) and others were investigated in Russian.

The data obtained indicate that the Zipf's constant for Russian works is within:

- 18th century from 0.037 to 0.064
- 19th century from 0.04 to 0.065
- 20th century from 0.021 to 0.05

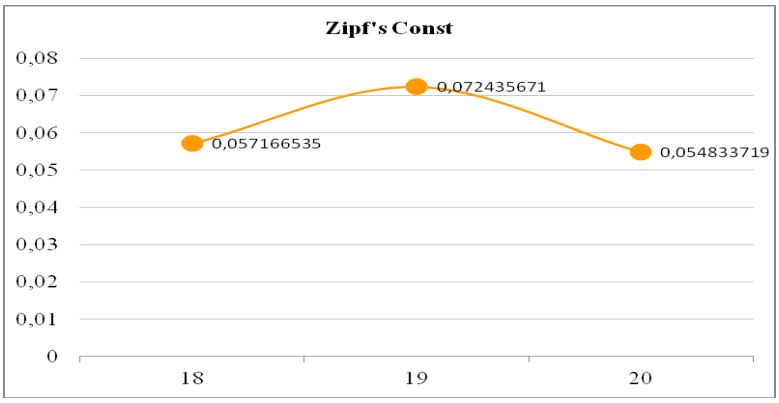

Figure 2: Counting Zipf's constant for Ukrainian.

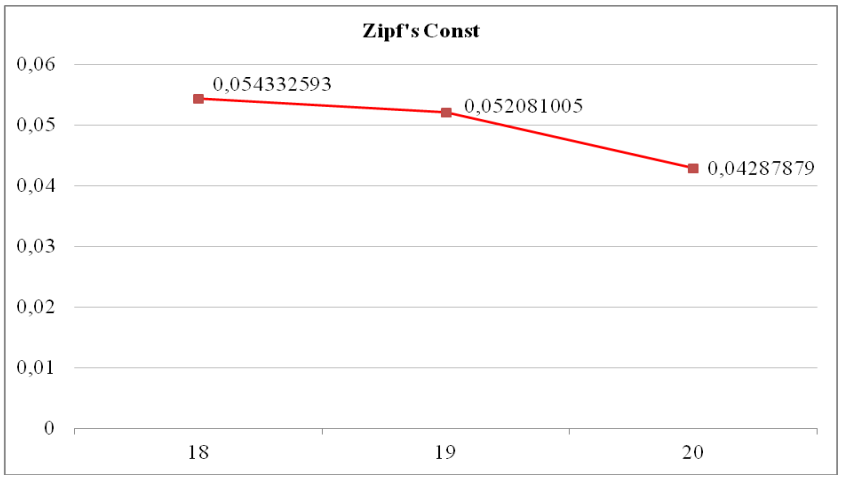

Figure 3: Counting Zipf's constant for Russian.

The results suggest that the language underwent an intense change in the 18th century. Perhaps it was the period of the formation of the language and its foundations. In the 19th century, language changes are minimal, i.e. the language was as stable as possible during this period of time. In the 20th century, the figure shows that the language undergoes an intense change. But given the fact that, for all centuries, the constant is approximately in the same ranges, we can assume that the identity of the language has remained unchanged for three centuries.

Also for comparison were used "Bogurodzica" (18th century), "Creations of Adam Asnyk" (19th century), "Creations of Tadeusz Boy-Żeleński" (20th century) ad other works in Polish. The obtained data can be seen in Figure 4.

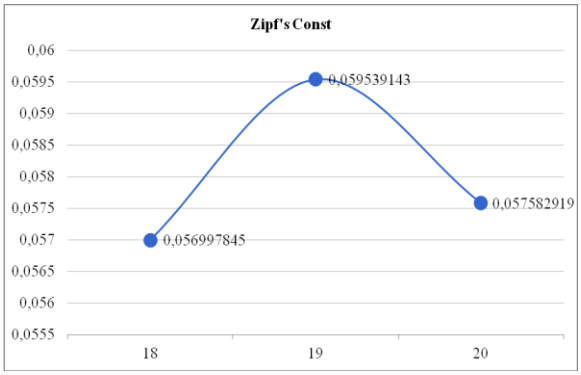

Figure 4: Counting Zipf's constant for Polish.

The data obtained indicate that the Zipf's constant for Polish works is within:

- 18th century from 0.047 to 0.067
- 19th century from 0.058 to 0.064
- 20th century from 0.054 to 0.068

The language has been subject to minimal changes over three centuries and its identity has remained almost unchanged.

4 CONCLUSIONS

Comparing the results, we can assume that the most steady and stable language with the maximum preserved identity among the languages we have chosen was Polish, and the most modified one with the preserved identity is Russian. The Ukrainian language underwent not only structural changes, but also changes in its identity (in the 19th century), but in the 20th century, the results testify to its stabilization and the identity to become close to Polish.

Regarding the practical relevance, this program can be used not only to check the language change over time, but also, for example, to check the authenticity of artistic translations, also to check the language features depending on the social sphere. Can be tracked changes in language depending on the region and migration of the population, which could be one of the points of our next research.

REFERENCES

[1] Á. Corral, G. Boleda, and R. Ferrer-i-Cancho, “Zipf’s Law for Word Frequencies: Word Forms versus

Lemmas in Long Texts", PLoS ONE, vol. 10 (7), 2015.

[2] M. Cristelli, M. Batty, and L. Pietronero, "There is More than a Power Law in Zipf", Scientific Report 2, no. 812, 2012.

[3] A. Saichev, Y. Malevergne, and D. Sornette, "Theory of Zipf 's law and beyond", Lecture Notes in Economics and Mathematical Systems 632, Springer, Heidelberg, Germany, 2010.

[4] V. Bochkarev and E. Lerner, "Calculation of Precise Constants in a Probability Model of Zipf's Law Generation and Asymptotics of Sums of Multinomial Coefficients", International Journal of Mathematics and Mathematical Sciences, vol. 17, 2017.

[5] M. Hauser, Ch. Yang, R. Berwick, I. Tattersall, M. Ryan, J. Watumull, N. Chomsky, and R. Lewontin, "The mystery of language evolution", Frontiers in Psychology, vol. 5, 2014.

[6] W. Fitch, "Empirical approaches to the study of language evolution", Psychonomic Bulletin & Review, vol. 24 (1), 2017, pp. 3-33.

[7] Y. Zhao and J. Zobel, "Search with style: Authorship attribution in classic literature", In Proceedings of the Thirtieth Australasian Computer Science Conference, Association for Computing Machinery, 2007.

[8] K. Shilova, D. Goncharenko, L. Bodnar, O. Britavska, and A. Grechkosiy, "Zipf's laws and translation approaches", Proc. of the 7th Intern, Conference "Information Technologies and Management", Riga, 2009, pp. 61-62.

[9] A. Kiv, D. Goncharenko, Ye. Sedov, L. Bodnar, and N. Yaremchuk, "Mathematical study of evolution of Russian language", Computer Modeling & New Technologies, vol. 12 (1), 2008, pp. 56-59.

[10] A. Kiv, L. Bodnar, O. Britavska, E. Sedov, N. Yaremchuk, and M. Yakovleva, "Quantitative analysis of translation texts", Computer Modeling & New Technologies, vol. 18 (12C), 2014, pp. 260-263.

[11] Analyzing and Interpreting Large Datasets. Atlanta, GA: Centers for Disease Control and Prevention (CDC), 2013.

[12] S. Bird, "Natural Language Processing with Python", O'Reilly Media Inc, 2009, p. 504.

[13] R. Dale, H. Moisl, and H. Somers, "Handbook of Natural Language Processing", Marcel Dekker, 2000.

[14] D. Mertz, "Text Processing in Python", Addison-Wesley, Boston, MA, 2003.