**UDC 004.021:81**
**Y.Levus[1], S.Buk[2], Y.Yavorskyi[1]**

[1] National University "Lviv Polytechnic"
Department of Software
[2] Ivan Franko Lviv National University,
Department of General Linguistics

# THE ALGORITHM OF SHOWING TEXT'S LEXICAL RICHNESS CHANGE

**Described is the algorithm of detecting changes in the ratio of different words to the total number of words in the text which can be used to address the issues of determining the author's style. The problem of comparing text styles works is relevant in both philological and historical studies, as well as in computer science. The use of these comparison methods can improve the quality of classification and text collections management, which is important for search engines and large repositories of text data.**

**A distinctive feature of the algorithm among similar ones is its ability to analyze the dynamics of lexical richness of the text. The algorithm is implemented in software system for texts analysis.**

**Keywords: algorithm, lexical richness, author's style, word.**

## Issues concerning the analysis of the text author's style

Today, a lot of attention in the field of applied linguistics is concerned about analyzing the text in terms of the author's style features used in it. The problem of comparing text styles is relevant in many areas of human activity. In historical studies, analysis of the style used to determine authorship or the making of historical documents, in philological disciplines - to explore the stylistic features of texts or language works in various genres, authors etc. In practice, the task of comparing text style challenges in criminology, for example, to establish an author by written threats or identify its individual characteristics during the operational-search measures.

Quantitative approaches to solving such problems are particularly relevant as they are compared to automate text styles give a formalized objective decision. The development of these approaches is also important for science because they can improve quality of classification and ordering of text collections which is important for search engines with large repositories of text data. Comparison of text styles produced is generally based on a set of attributes that reflect the style of text properties. Usually considered are frequency characteristics (frequency of certain words, letter combinations, etc.) that can be easily formalized for their help with quantitative (frequency) analysis in [1].

Despite the large number of approaches to solving the problem of establishing the author's style, there are still unexplored methods for its numerical characteristics. In particular, there are no algorithms that could calculate the change of lexical richness. Lexical richness (dictionary diversity, lexical density [2]) - the ratio of different words to the total number of words in the text, which is used as a quantity that characterizes the text (term used here is "lexical richness" because existing naming conventions differ in essence - they strictly used the total number of words in the text [2], and in this paper we used the number of words in the selected text block). It is therefore advisable to develop an algorithm to use the results of the calculation as additional parameters author's style. Graphical representation of intensity changes in the text of a large volume will conclude particular author's style.

## Existing methods for solving

In [3] the lexical richness of texts is being used as an auxiliary number of psycholinguistic analysis of texts, and appears only as a single number for each text. Also this value is used as an auxiliary value in the search for information [2].

There are two functionally non-completed software tools that implement the task of the lexical richness of the text. Both products have a web interface.

*Wordcounter* [4] handles texts in English only, functions are: printing a list of most commonly used words and tuning of basic settings. However, a significant drawback is that the comparison of words is not using the infinitive, but the base, which is calculated simply by discarding the last few letters. Therefore, words that have replaced letters in morphing do not match.

*Document Information Tool* [5] also handles text only in English, provides output statistics of words and letters. But the analysis does not take into account the words of the same infinitive form, they are different.

## Problem

It is necessary to develop and implement a mapping algorithm that displays change of the lexical richness of the text. In particular, the implementation should have support for dictionaries of different languages for universality.

## Decision

Among available in the public domain dictionaries of word forms dictionary format Hunspell [6] was selected, since in this format you can find dictionaries for most modern languages, including Ukrainian. Obviously, to calculate the richness of text we need to convert entire text to speech infinitive form, and then calculate the fraction obtained by dividing the number of different words in the formed array by the total length of the array.

The general format of the data in the Hunspell dictionaries:

word/abc,

where abc – list of classes, according to which words can be formed from the word-form (transformation rules are contained in the supplementary affix file).

To improve the performance of transforming dictionaries into array of vocabulary forms, during the operation initialization it is needed to generate all possible forms for each word.

Thus, finding the saturation algorithm for text block is as follows:

1) Initialization the dictionary
2) Separation of the input text to speech
3) Conversion of each word input text in its dictionary form, words that are unknown program dismissed (this approach is minimally distorting influence of unknown words in the test result)
4) Calculation of Np. - The number of distinct words in the result set vocabulary of forms
5) Calculation of saturation - Np. / Nzah.

The resulting saturation calculation function is used to detect changes in richness during the writing of the text by the author.

Solving this problem is based on the selection of a particular group of words (Fig. 1).

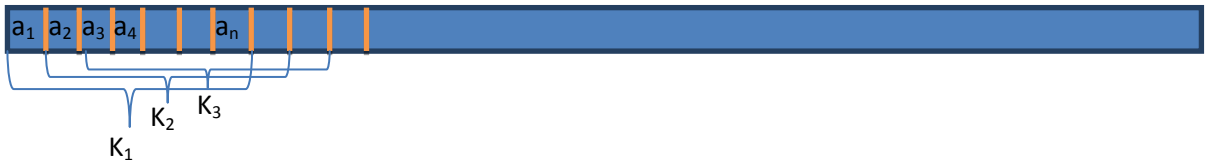



*Fig. 1. Calculation of change in lexical richness.*

Here $\mathbf{a_i}$ – a word from text; $\mathbf{K_j}$ – evaluated value of richness for j's group of words with length **n,** $j = \overline{1, N}$ . In fact, there is a movement of block, which is responsible for the beginning and end of the text for the calculation of richness, one word for each new point graph (array of values). Therefore, the number of points on the resulting graph will be $D = N - n + 1$.

Algorithm for evaluating richness change:

1) Obtaining a block of the first n words from converted text words to dictionary forms
2) Calculating the richness of the resulting block and saving it in the resulting array
3) If the last word of the block - this is the last word of the converted text, then exit procedure
4) Discard the first word and go to the step 1)

For ease of interpretation of the text analysis results text analysis experts should use a graph of lexical richness block on the position in the text.

After analyzing several works for different values of n, it was determined that the analysis of graphical output should be performed when n = 500. With much larger or much smaller value the graph is smoothed, as the difference in intensity is minimal (for large values of n it is always low, and for small values of n - always high).

The proposed algorithm has been implemented in the software MorphAnalyzer (Fig. 2-5) using environment Eclipse IDE with plugin WindowBuilder [7]. Window interface was built using JFace + SWT [8]. The principles of object-oriented programming and modular programming were applied.

MorphAnalyzer software allows you to analyze the input text and display some quantitative, frequency and other information about it. When you select dictionary files and click "Initialize" dictionaries are being loaded into memory.

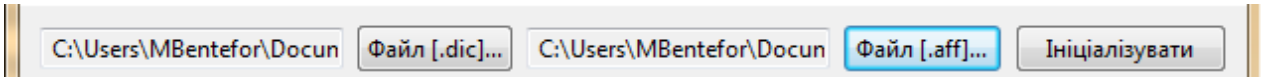



*Fig. 2. Selecting files and initializing dictionary.*

Depending on the operating system, this procedure can take up to 30 seconds. Upon completion, the program will display information about received dictionary data.

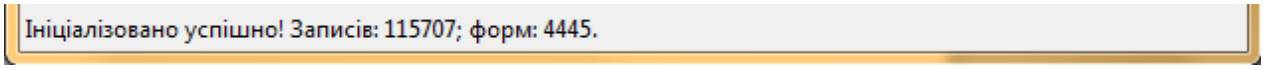



*Fig. 3. Result of initialization.*

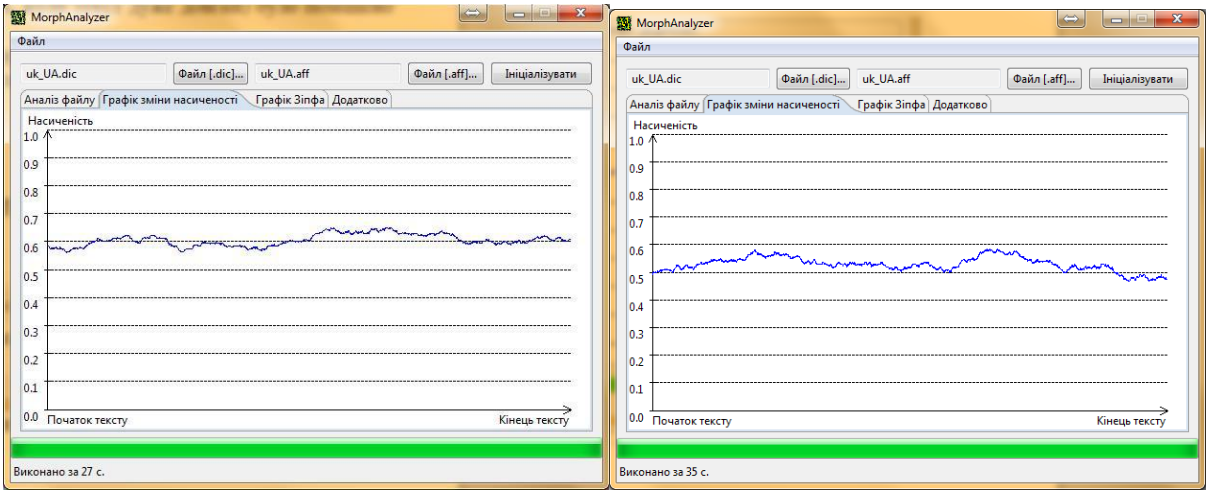
You must select the input text file and click "Analyze" to process texts. Time costs for the operation of analysis depends on the size of the input file. In large-volume data analysis process can take up to few hours (Table 1).

*TABLE 1. The analysis of texts*

| Text Size (Words) | Analysis of execution time |
|---|---|
| 1129 (science post) | 36 s |
| 3534 (article) | 77 s |
| 21650 (story) | 5 m 31 s |
| 225311 (novel) | 43 m |

Testing was conducted on a working machine with a processor Intel Core i7 (4 cores, 2.4 GHz) running Windows 7.

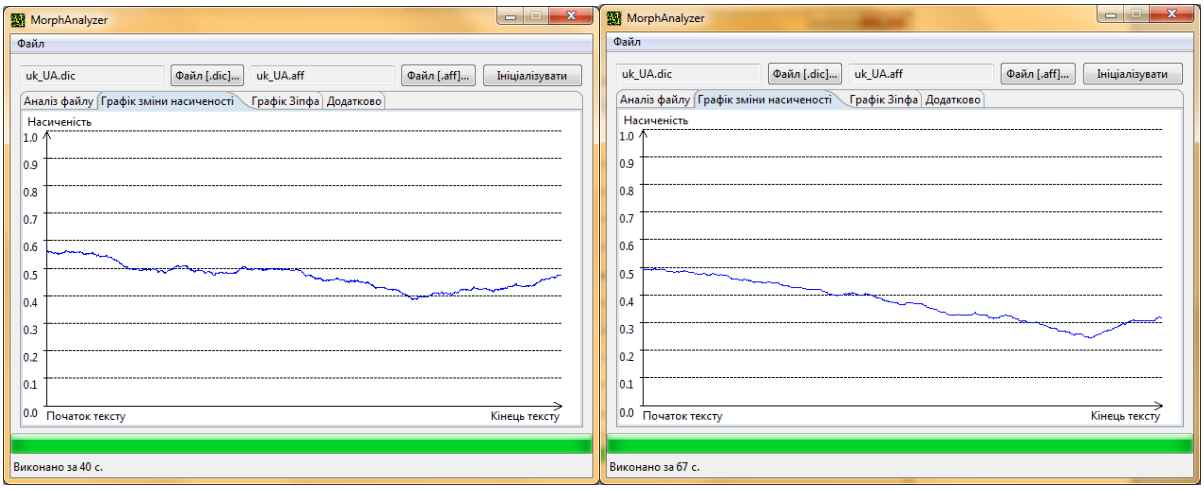
As the conducted numerical experiments, text style determines the nature of the resulting graph. The analysis of artworks is shown in Fig. 4.




a) b)

*Fig. 4. a - the result of analysis of the work of Shevchenko's "In all their own fate";*
*b - Result of analysis of work "Book of Veles"*

The large number of repetitions in texts of a scientific nature manifests itself in a general descent of graph (Fig. 5).




a) b)

*Fig. 5. a - the result of analysis of scientific articles for the IT industry;*
*b - Result of analysis of scientific articles for the electronics industry.*

Thus, the use of software algorithms can be used for further linguistic and psycholinguistic research.

## Conclusions

This paper discusses the relevance of the determination of author's style in various fields of research and the algorithm that provides a function to solve such problems.

The algorithm is implemented in a software application whose main function is to display graph of the lexical richness change. Using the developed software one can find common features in the works of same author, i.e. getting a new characterization of author's style. For example, after analyzing a small number (~10) prose works of Ivan Franco, it was concluded that, unlike other authors, lexical richness in his novels varies quite widely.

The results of analysis show promise in the use of the algorithm applied in many industries that use psycholinguistic and linguistic analysis of texts [9-12]. It is planned to improve an existing software implementation with the development and implementation of new algorithms to automate the comparison of author's styles.

*1. Shevelev O.G. /Development and analysis of algorithms for comparing styles of textual works: thesis abstract. – Tomsk. – 2006. 2. Veres M.M., Lemkivskyi Y.O., Omelchenko O.A. /Mass-distributed search robot //Problems of Information Technologies. – 2011. – №1 (009). 3. Psycholinguistic word analysis - Wikipedia, the free encyclopedia [electronic resource]. – Web page: http://uk.wikipedia.org/wiki/Психолінгвістичний_текстовий_аналіз (2013). 4. Wordcounter [Electronic resource]. – Web page: http://www.wordcounter.com/ (2004). 5. Character And Word Counter With Frequency Statistics Calculator [Electronic resource]. – Web page: http://www.csgnetwork.com/documentanalystcalc.html (2013). 6. Man hunspell – format of Hunspell dictionaries and affix files [Electronic resource]. – Web page: http://pwet.fr/man/linux/fichiers_speciaux/hunspell (2013). 7. Eclipse Workbench User Guide [Electronic resource]. – Web page: http://help.eclipse.org/juno/index.jsp?nav=%2F0 (2012). 8. JFace Eclipse toolkit [Electronic resource]. – Web page: http://wiki.eclipse.org/index.php/JFace (2013). 9. Kishtimova I.M. / Psychosemiotic text analysis: the diagnostic value of the category "time". – Web page: http://www.lib.tsu.ru/mminfo/000085170/26/image/26-050.pdf. 10. Psychology - Wikipedia - the free encyclopedia [electronic resource]. – Web page: http://ru.wikipedia.org/wiki/Психология (2013). 11. Gorelov I. N., Sedov C. F. /Fundamentals of psycholinguistics. — Moscow. – 1997. 12. Zasekina L. V., Zasekin S.V. / Introduction to Psycholinguistics – Ostrog: National Publishing House. Univ "Ostrog Academy", 2002. – 168 c.*