Deep Learning-Based Analysis of Ancient Greek Literary Texts in English Version: A Statistical Model Based on Word Frequency and Noise Probability for the Classification of Texts

Zoltan Gal, and Erzsébet Tóth

***Abstract*—In our paper we intend to present a methodology that we elaborated for clustering texts based on the word frequency in the English translations of selected old Greek texts. We used the classification system of the ancient Library of Alexandria, devised by the prominent Greek scholar-poet, Callimachus in the 3rd century BC., as a basis for categorizing literary masterpieces. In our content analysis, we could determine a triplet of a, b, c values for describing a power function that appropriately fits a curve determined by the word frequencies in the texts. In addition, we have discovered 16 special features of the different texts that correspond to various token categories investigated in each text, such as part of speech of the word in the context, numerals, subordinate conjunction, symbols, etc. We have developed a cognitive model in which several hundred different subtexts were utilized for supervised learning with the aim of subtext class recognition. Concerning 200 subtexts, the triplet of a, b, c values, the classes of the subtexts, and their 16-dimensional feature vectors were learnt for the Recurrent Neural Network (RNN). It turned out that the Long-Short Term Memory RNN could efficiently predict which class a chosen subtext could be categorized into without considering the interpretation of the content. The influence of the non-zero error rate of new communication services on the meaning of the transferred texts was also investigated. The impact of the noise on the classification accuracy was found to be linear, dependent on the character error rate**

***Index Terms*—deep learning, old Greek literary texts, Pinakes, automatic content analysis, text classification, Recurrent Neural Network (RNN), Long-Short Term Memory, noisy texts.**

I. Introduction

In the growing methodological pool of modern social science, content analysis is still a frequently used method [1] and one of the most significant research techniques [2, 3] applicable to quantitative and qualitative data. This method allows verbal, written, and visual data to be thoroughly investigated, systematically analyzed, and categorized. In the Internet era, the use of social media platforms is becoming inevitable for the members of the generation Cognitive Entities (CE). Content analysis can also be used for examining networked short texts [4]. There is no doubt that the careful reading and manual processing of these texts is a very time-consuming task for researchers. For this reason, automatic content analysis is applied to predict users' positive or negative preferences, and interests based on their short messages on social media. This new type of content analysis which makes possible the classification of users' feelings e.g. in the case of IMDB movie film reviews is called sentiment analysis in the field of deep learning [5]. The role of artificial intelligence and machine learning has become more and more important currently in the extraction of meaningful information and knowledge from different types of Big Data sources [14, 15].

Computer software could be used effectively at two different stages of textual content analysis: first, for storing, examining, and reporting research data; second, for automatic screening of texts, identifying and coding words and expressions [6]. An algorithmic solution could be applied in six steps: i) splitting texts into various segments; ii) determining similarities; iii) clustering; iv) cluster labeling; v) examining categories; vi) presentation of results in detail [7]. In these text analyses supervised or unsupervised automatic classification could be utilized. The researcher has to modify and prepare the texts for automatic content analysis. The preparation phase is composed of gathering textual data from social media or networks, preprocessing the texts, and defining research aims [4].

In section two a short overview of the related works is presented in the context of virtual library projects. Section three describes the analysis methodology accomplished on 37 ancient Greek texts translated into English. After highlighting the basic statistical properties of those texts, their preprocessing and neural network-based evaluation are discussed. Section four concludes and recommends the possible continuation of this research topic.

II. Related Works

The famous Great Library of Alexandria gathered all the available copies of the authentic old Greek literary texts from ancient times. In this respect, it could be regarded as a universal library representing a frequently cited symbol of human knowledge till nowadays. In its collection, the ancient texts were arranged in the alphabetical order of author names and

Z. Gal is with Faculty of Informatics, University of Debrecen, Debrecen, Hungary (e-mail: gal.zoltan@inf.unideb.hu)

E. Tóth is with Faculty of Informatics, University of Debrecen, Debrecen, Hungary (e-mail: toth.erzsebet@inf.unideb.hu).

DOI: 10.36244/ICJ.2024.5.1

classified according to their literary genres based on the Pinakes compiled by the illustrious Greek scholar, Callimachus in the 3rd century BC. In the framework of the cognitive infocommunications (CogInfoCom) research [8, 9, 10], a virtual library project was launched in 2013 to develop the three-dimensional virtual library model (3DVLM) of the ancient Library of Alexandria by collecting the relevant verbal and multimedia content about it [11]. Later this project continued to provide an overview of the greatest achievements of Callimachus by presenting the web content about his life and literary works in carefully arranged smartboards in the 3D Castle space of the MaxWhere Seminar System [12, 13].

The famous paper demonstrates that the source language of medium-length speeches in the EUROPARL corpus can often be identified through frequency counts of word n-grams, achieving an accuracy between 87.2% and 96.7% depending on the classification method. The study delves into identifying powerful positive markers and examining linguistic, cultural, and domain-related aspects. When considering all six target language versions, classification is more accurate compared to situations with only a single version available. The research also explores the diverse nature of strong markers, encompassing vocabulary, discourse structure, syntax, and contrasts between source and target languages, emphasizing the need for harmonization in terminology for improved information retrieval in parliamentary proceedings. The paper highlights the importance of further research to understand translation processes, automation of target and source language phrase retrieval, and the potential incorporation of source language considerations in EUROPARL translation models [18].

In their paper, prominent authors outline experiments in fine-grained stylometric analysis for distinguishing features among contemporaneous literary translations, both parallel and non-parallel, of works by the same author. Focusing initially on translations of Henrik Ibsen's drama “Ghosts”, the study extends to explore prose translations of Anton Chekhov by different translators, such as Marian Fell and Constance Garnett. Various machine learning approaches, including Support Vector Machines and Decision Tree classifiers, are employed to identify textual features, comparing their frequencies in translations to those in reference corpora. The study successfully establishes that common word unigrams and bigrams serve as salient features for classifying translator fingerprints, achieving accuracy measurements exceeding 90%, and reveals distinctive stylistic traits between translations by William Archer and R. Farquharson Sharp of Ibsen. Cross-validation experiments suggest the effectiveness of both document-level features and n-grams in distinguishing translators, with clustering experiments showcasing differences in text clustering based on frequent words versus discriminating words identified through supervised machine learning [19].

A frequently cited study explores differences between translated and non-translated texts, acknowledging two sources of variation: interference from the source language and general effects of the translation process. The authors identify consistent effects across texts translated from the same source language through text categorization experiments, indicating a continuum between source-language-specific and general translation effects. Remarkably, classifiers based solely on function words can accurately distinguish translated from original texts, even across unrelated source languages and multiple genres. The findings suggest that both the source language and the act of translation significantly influence the characteristics of translated texts, and the observed differences align with linguistic typology, offering insights into language distances and potential applications in computational tasks, such as improving machine translation and constructing language models [20].

III. Analysis Methodology

Ancient texts, like modern texts, describe a sequence of events in chronological order, even in multiple time planes. A time plane is a story happening in a dedicated geographical area in a limited time interval. It is a crucial time structure that provides logical connections to other parts of the literary work, offering a convergent evolution of the whole story. The author of the text uses his style in the narration of the story. This fact generates the cohesion of that text, which can be used in the automatic characterization of each subtext belonging to the main text. The analysis methodology presented in the following proves a strong correlation between the patterns of part of speech frequencies and the classes of texts.

Kaggle Database contains 1/3 million most frequent English Words on the Web [16]. The relative frequency of the words versus rank is represented in Fig. 1. Rank in this context means an order of the number of words in the analyzed text. The fitting equation of the intensity curve we found to be given the following formula:

$$y(x) = exp(\alpha \cdot x^{\beta} + \gamma) \quad (1)$$

where y is the number of items, and x is the rank of items corresponding to Fig. 1. Parameter triplet (α, β, γ) characterizes the modern English language used in the Internet.

We note that this list includes stopwords of the English language, which radically influence the curve at the top ranks. In most computational stylistics papers, stopwords are retained to reveal the style, but in our case, we investigated only the significant words describing the specific theme. By removing stopwords, short words ($|w| \geq 1$), and long words ($|w| \leq 15$) from the English texts, which are later called unnecessary words, the relative frequency of the remaining words changes depending on the context of the text (see *GenerateParams* in subsection B*)*. The term $|w|$ represents the length of words in the number of characters.

A. Characteristics of the Processed Texts

We downloaded 37 ancient texts from Project Gutenberg (see Table I). These texts were split into two main classes (poets and prosaists) and an additional six subclasses devised by Callimachus according to the literary genres of the texts, e.g. tragedians, comic playwrights, speakers, etc. So that the essential content of the texts could be examined, we had to remove unnecessary words and punctuation marks from the original texts.

Word intensity in the merged text (T37) of the analyzed 37 texts is represented in Fig. 2. The fitting equation of the intensity curve was found to conform to the formula below:

$$y(x) = a \cdot x^b + c \qquad (2)$$

where y is the number of items, and x is the rank of items. Equations (1) and (2) are very different. The former equation is exponential to a power function and the latter is a power function. The parameter triplet (a, b, c) characterizes the special features of the ancient English language.

TABLE I.
CLASSES OF ANALYZED ANCIENT TEXTS

	Callimachus' Class	No. of Texts
1.	Poets - Tragedians	8
2.	Poets - Comic playwrights	8
3.	Poets - Epic poets	11
4.	Poets - Lyrical poets	2
5.	Prosaists - Philosophers	5
6.	Prosaists - Speakers	3

Looking at Fig. 1 and 2, we can conclude that modern English tends to use more intensively general words. Those insignificant terms are called stopwords in machine learning-based analysis. Rijsbergen calls attention to Luhn's oeuvre in automatic text analysis, who supposed that word frequencies can be applied to extract words and sentences to represent the content of a document. Let f be the word frequency in a particular position of text and r the rank order of the words (i.e. the order of their frequency of occurrence), then a plot f versus r produces a hyperbolic curve. In addition, this curve presents Zipf's law, which says that the product of word frequencies and their rank order is approximately constant. Luhn applied this law as a null hypothesis to determine the upper and lower cut-offs of the rank order of words. He meant by the significance of the words, their ability to discriminate the topic or content of the text. Using these arbitrarily specified cut-offs, he excluded insignificant words, such as common and rare words, from the rank order of items, and thus he could find those significant words that describe the content of the text [17].

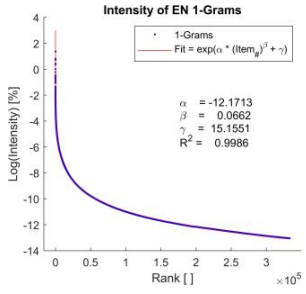


Fig. 1. Frequency of the English words in the Internet texts including all words (logarithmic scale) (Source of raw data: [16])

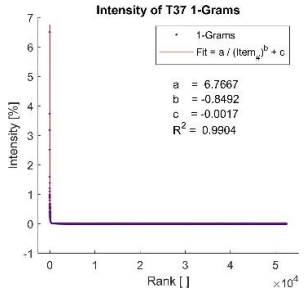


Fig. 2. Frequency of the English words in the analyzed 37 ancient texts including all words (linear scale)

High-level programming languages offer advantageous procedures to eliminate such insignificant elements after converting all the text strings into a list of tokens. Tokens in this context are words or punctuation marks, as well. Examples of such cleaned texts are represented in Fig. 3 (Text1: original author: Aeschylus, translated by Murray, Gilbert, title: Agamemnon, Callimachus' class: Poets – Tragedians; Text4: original author: Apollonius Rhodius, translated by Seaton, R.C., title: The Argonautica, Callimachus' class: Poets - Epic poets). Because of spatial limits in the graphs, just a few elements with the highest ranks are listed from the top list of the tokens. The same methodology of graphical visualization is applied for 2-Grams in Fig. 4, too.

Fig. 3. Word cloud of cleaned Text1 (left) and cleaned Text4 (right).

Stopwords, short words, and long words were removed except archaic English words (i.e. thou, hath, thy, etc.). These cleaned texts contain theme-specific words on the top of the word ranks (see Fig. 3). Word cloud is a representation mode of the token frequency ranks. The larger the size of the characters is, the higher the frequency of the tokens in the text. N-Gram is a group of N tokens ($N = 1, 2, \ldots$) located next to each other in the text. 2-Grams are neighboring word pairs in a fixed order of occurrence. Such examples of 2-Grams for the ancient texts mentioned above are represented in Fig. 4.

Fig. 4. Bag cloud of 2-Grams of Text1 (left) and Text4 (right).

To execute the removal of the unnecessary tokens from the texts and evaluate artificial intelligence-based procedures, we used embedded packages of the Matlab system: Text Analysis Toolbox and Machine Learning and Deep Learning Toolbox. A list of modern English stopwords we use is integrated into the actual version of the software.

B. Preprocessing of the Texts

To eliminate the effect of stopwords and to generate the rank-fitting parameters for each of $k = 37$ texts, we implemented a cleaning and fitting algorithm given in the following meta code sequence:

```
[a, b, c] = GenerateParams(text1,..., textk):
    global doc, tokenDetails
    for textid = 1:37:
      for partid = 1:10:
        doc = Import(textid, partid)
        doc = Lower(doc)
        doc = TokenizeDoc(doc)
        doc = AddPartOfSpeechDetails(doc)
        tokenDetails = TokenDetails(doc)
        doc = NormalizeWords(doc)
        doc = RemoveStopWords(doc)
        doc = ErasePunctuation(doc)
        doc = RemoveShortWords(doc, max = 2)
        doc = RemoveLongWords(doc, min = 15)
        bag = BagOfWords(doc)
        topBag = TopkWords(bag,noTokens = 1000)
        [a(textid,partid), b(textid,partid),  c(textid,partid)] =
        FitCurve(topBag, type = 'hyperbole')
# end GenerateParams()
```

An explanation of tasks executed by each function is enumerated in Table I. The majority of the procedures listed are embedded methods of the Matlab software.

TABLE II.
FUNCTIONS OF TEXT PROCESSING

	Function Name	Effect
1.	*Import()*	Import text part from text
2.	Lower()	Convert string to lower cases
3.	*TokenizeDoc()*	Convert string into tokens
4.	*AddPartOfSpeechDetails()*	Add part of speech details
5.	*NormalizeWords()*	Reduce each token to stem
6.	*RemoveStopWords()*	Delete stopwords of the modern English language
7.	*ErasePunctuation()*	Delete punctuation tokens
8.	*RemoveShortWords()*	Delete short tokens
9.	*RemoveLongWords()*	Delete long tokens
10.	*BagOfWords()*	Generate a bag of words
11.	*TopkWords()*	Generate a top list of words
12.	*FitCurve()*	Fit hyperbole curve to top list

Elements of the preprocessing list were created based on a logical sequence of the tasks. After importing and converting to lower cases, the following steps were executed: i) *Tokenize the text*: This is the process of breaking down a text into individual words or tokens. This step helps in converting the raw text into a structured format where each word is a separate element. It is a fundamental step in text processing. ii) *Add part of speech details*: Knowing the part of speech of each word can provide valuable linguistic information. This information can be useful in later stages of analysis, such as when identifying key terms or relationships between words. iii) *Normalize words*: Stemming involves reducing words to their base or root form. This step helps in consolidating different forms of the same word, which can improve the efficiency of downstream processes by treating variations of a word as a single entity. iv) *Remove a list of stopwords*: Stopwords are common words (e.g., "and," "of," and "the") that often do not carry much meaning and can be considered noise in text analysis. Removing stopwords helps focus on the more meaningful terms in the text. v) *Erase punctuation*: Punctuation marks usually do not contribute much to the semantic meaning of the text and can introduce noise. Removing punctuation simplifies the text and ensures the focus remains on the words. vi) *Remove short words* (with 2 or fewer characters): Very short words, often with two or fewer characters, might not carry significant meaning and can be removed to reduce noise in the data. vii) *Remove long words* (with 15 or more characters): Extremely long words might be outliers or noise in the data. Removing such long words can improve the efficiency of subsequent analysis. viii) *Generate a bag of words*: A bag-of-words representation converts the preprocessed text into a numerical format, representing the frequency of each term in the document. It is a common representation used in many NLP tasks. ix) *Generate a top list of words*: Creating a top list of words helps in identifying the most frequent and potentially important terms in the text. This list can be used for further analysis or visualization. x) *Curve fitting* generates feature triplets (a, b, c) as a compressed property of the text (or subtext).

Note that the function AddPartOfSpeechDetails() retokenizes the text for part-of-speech tagging. This is the reason why it is executed before the NormalizeWords() and stopwords filtering tasks. Stopwords extraction takes place after lemmatization to reduce better the remaining number of words in the dictionary forms. These steps allow tokens to remain in cohesion in the created bag of words (see the order of steps 4, …, 9 in Table II). In addition, Fig. 5 provides further details about the size of the original and cleaned texts. Note that the effect of cleaning tasks results in decreasing text length by one order of magnitude.

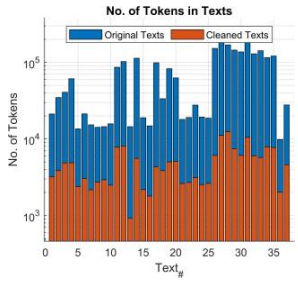


Fig. 5. Reduction of the text length after the preprocessing.

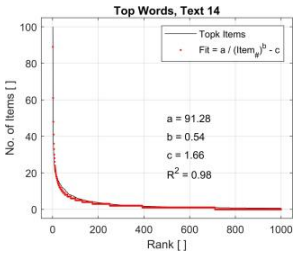


Fig. 6. Number of words in decreasing order of occurrence in Text14.

The resulting two-dimensional arrays a, b, c contain fitting parameters of the text part identified by *textid(1:37)* and *partid(1:10)* indexes, respectively. The function *FitCurve()* executes regression of the rank of the top 1000 words to the hyperbolic equation below:

$$y(x) = \frac{a}{x^b} - c \tag{3}$$

where y is the number of items, and x is the rank of items. Formula (2) and (3) are similar, suggesting no modification of the equation when consistent parts of the text are analyzed. Each parameter triplet *(a, b, c)* characterizes the relative number of words in the corresponding text quantitatively. Variables *doc* and *tokenDetails* have a global scope in the main program and are used for other processing.

An example of the fitting is illustrated in Fig. 6. This text belongs to the Prosaists – Philosophers class. Note that because we have $(b, c) \neq (1,0)$, hence the hyperbole is not a symmetrical curve. We found that all the fitting parameters (b, c) of the 37 texts conforming to the relation (3) are different from $(0, 1)$, respectively. The log-log scale plot of the token intensities vs. ranks is given in Fig. 7. Note that these curves intersect each other in rare cases, so each curve suggests a significant feature of the corresponding text.

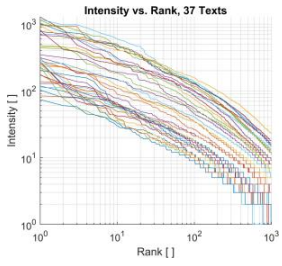


Fig. 7. Log-log plot of the number of words in decreasing order of occurrence in each text.

Representation of the 37 ancient texts in the space of fitting parameters (SFP) can be seen in Fig. 8.

$$SFP = \{(a, b, c) |\, a, b, c, : text\ fit\ parameters\} \quad (4)$$

We realized that texts belonging to the same Callimachus' class are located in compact regions of the SFP.

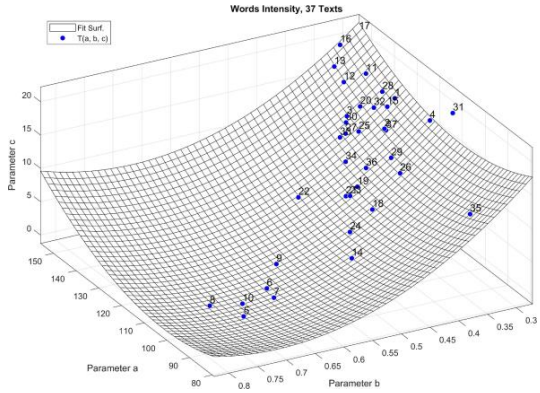


Fig. 8. Location of the texts in the 3D space of parameters (a, b, c).

Some regions on the surface have a small range; others have a more extensive range of parameters. We mention that the region of texts 5-9 belongs to the class 'Poets - Comic playwrights', and that of texts 27-35 belongs to the class 'Poets - Epic poets'. Not all the texts are located homogeneously in this parametric space SFP (i.e. Text35 in the right part of the surface).

We formulate a hypothesis that by dividing each text into ten subtexts, their placement in the parametric space SFP is not changing radically. To validate the hypothesis, we generated $m = 10$ subtexts from each text, and we executed a cleaning and fitting algorithm on each element of the set of $37 \cdot m = 370$ subtexts. The size of each subtext is the tenth part of the parent text, providing 37 different sizes for 370 subtexts.

Fig. 9 proves this statement, but a few subtexts became far positioned in the space SFP (i.e. the top two subtexts). A possible reason for this divergence is that the tenth subtext of some texts contains footnotes generated by the translator, producing an exception of the subtext cohesion.

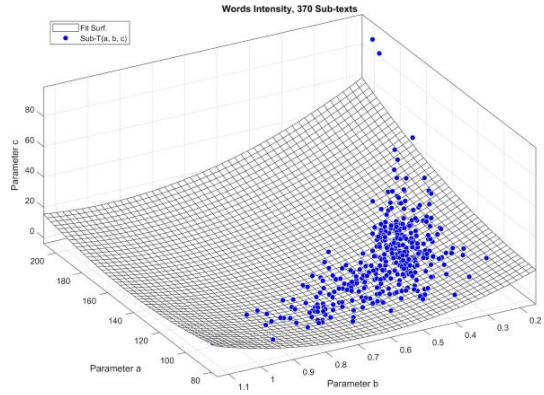


Fig. 9. Location of the subtexts in the space ST of parameters (a, b, c).

We found that the 37 texts and even 370 subtexts are located on a quadrature 'Poly22' type surface given by the following formula:

$$c = p_{00} + p_{10} \cdot a + p_{01} \cdot b + p_{20} \cdot a^2 + p_{11} \cdot a \cdot b + p_{02} \cdot b^2 \quad (5)$$

The values of the parameters $p_{ij}, i, j \in \{0, 1, 2\}$ and the coefficient of determination (R^2) of the fitting are given in Table III.

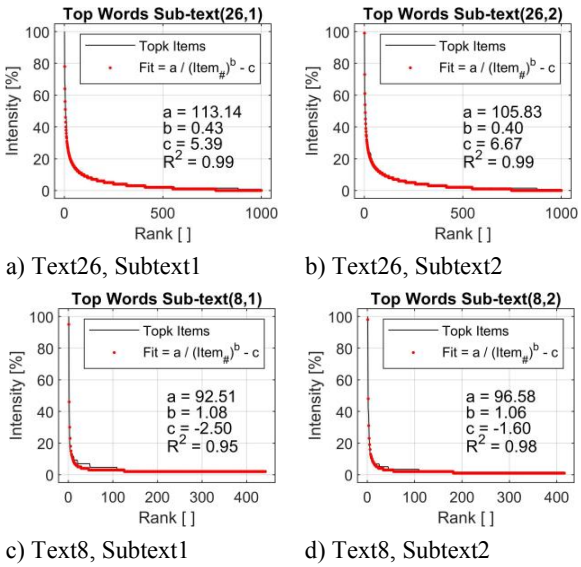
TABLE III.
PARAMETER VALUES OF SURFACE FITTING

Case	p_{00}	p_{10}	p_{01}	p_{20}	p_{11}	p_{02}	R^2
37 Texts	84.14	-0.75	-132.80	0.003	0.11	86.63	0.98
370 Subtexts	36.91	-0.22	-72.30	0.002	-0.30	71.65	0.97

Division of each text into ten subtexts has the following effect: the ranges of variable a remain the same for 37 texts and 370 subtexts: $a \in (80,150)$. In the case of 370 subtexts, the range for variables b and c increases: from $(0.3, 0.8)$ to $(0.3, 0.95)$ and from $(0, 20)$ to $(0, 80)$, respectively.

Examples of hyperbole fitting with a very good coefficient of determination are represented in Fig. 10. We mention that $R^2 > 0.95$ in each case of 370 subtexts. Subtexts 1 and 2 of texts 26 and 8 belong to different Callimachus' classes: Text26: original author: Homer, translated by Butler, Samuel, title: The Iliad of Homer, Callimachus' class: Poets - Epic poets; Text8: original author: Aristophanes, translated by an unknown person, title: The Frogs, Callimachus' class: Poets - Comic playwrights. We note that the rank range of items in subtexts is just 450 because of the reduced size of subtexts. This number is 1000 items for the texts. Another finding is the strong similarity of the subtexts' frequency curves belonging to the same text

(subtext(26,1) to subtext(26,2) and subtext(8,1) to subtext(8,2)).



a) Text26, Subtext1 b) Text26, Subtext2

c) Text8, Subtext1 d) Text8, Subtext2

Fig. 10. Top words of Text26 and Text8.

Due to the function *AddPartOfSpeechDetails()* (see Table II) we generated *tokenDetails* objects of each cleaned subtext. This object made it possible for us to have each token category (see Table IV). We generated a relative number of each token category for every subtext producing their histogram.

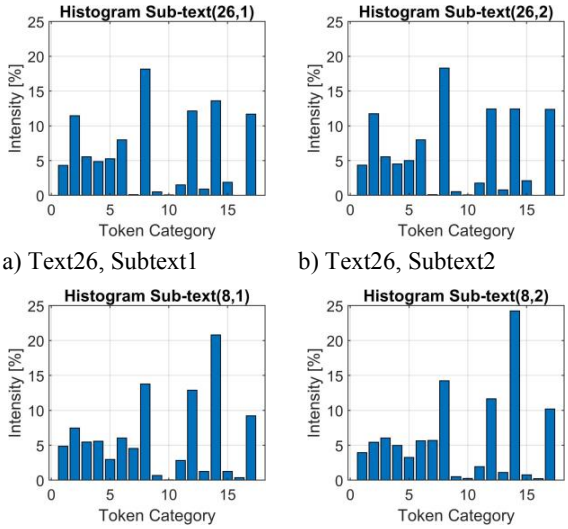
TABLE IV.
TOKEN CATEGORIES AS FEATURES OF THE SUBTEXTS

ID	Token Category	ID	Token Category
1	adjective	9	numeral
2	adposition	10	particle
3	adverb	11	pronoun
4	auxiliary-verb	12	proper-noun
5	coord-conjunction	13	punctuation
6	determiner	14	subord-conjunction
7	interjection	15	symbol
8	noun	16	verb

Histograms of the token categories for four different subtexts are given in Fig. 11. These values are normalized, giving the sum 100%. The value of each token category was found to be in the range $(0, 25\%)$ independently of the subtexts class.

The seventeenth category 'other' is not considered because it is a linear combination of the sixteen categories. We mention that some categories have very low intensity (e.g. symbol, numeral, and particle), and others have high values (e.g. noun and subord-conjunction). Others (e.g. adposition and interjection) have significant fluctuation, providing a difference between the features of the subtexts.

Features are represented by the relative histograms of the token categories. The correlation matrix plots in Fig. 12 and Fig. 13 are composed of cells of 10x10 and 100x100 pixels, respectively, forming squares and long lines (rows and columns) with homogenous colors. White cells on the main diagonal have a value close to 1 and prove the strong correlation between the subtexts of the same text.



a) Text26, Subtext1 b) Text26, Subtext2

c) Text8, Subtext1 d) Text8, Subtext2

Fig. 11. Histogram of token categories of Text26 and Text8.

Dark rectangles or line regions belong to those pairs of texts that are put into different Callimachus' classes. It can be observed in Fig. 12 (in the case of $m = 10$) that the range of the correlation matrix is [0.6, 1], and the mean of the elements is 0.9, proving the strong cohesion of the subtexts in general. The lower correlation is due to the reduced number of tokens of the compared subtexts. In the case of $m = 100$, the range has a lower minimum value, proving that shorter subtexts will not resemble each other. An extreme case is when subtexts contain just one token, and different tokens should belong to different classes, implying a low correlation.

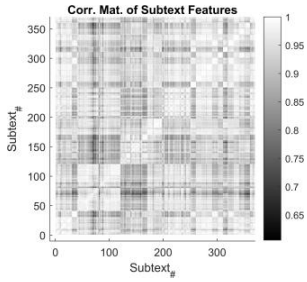


Fig. 12. Correlation matrix of subtext features (m = 10).

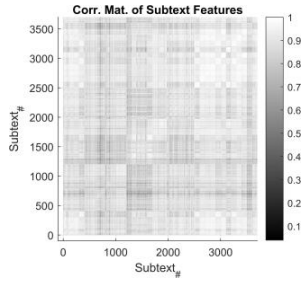


Fig. 13. Correlation matrix of subtext features (m = 100).

C. Neural Network-Based Processing of the Subtexts

Because of the relatively small number of analyzed texts, we divided them into subtexts. We used a significant property of these documents, namely the frequency distribution of words to characterize the texts. We have found that the subtexts belonging to a given text show similar quantitative behavior according to this property. In such an approach, we have $37x10 = 370$ subtexts, each characterized by a feature vector with 16 dimensions. These subtexts are grouped into six

Callimachus' classes according to literary genres. 200, 85, and 85 different subtexts were used for supervised learning with the aim of learning, validation, and testing, respectively. The architecture of the Recurrent Neural Network (RNN) is shown in Fig. 14.

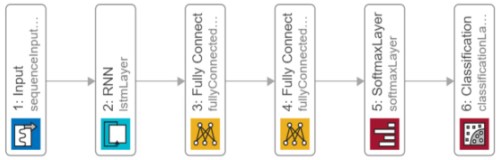


Fig. 14. Architecture of the applied RNN.

The RNN type was Long-Short Term Memory (LSTM), and the applied parameters corresponded to Table V. The LSTM layer (see layer 2 in Fig. 14) was used because RNNs, in general, are more stable in the classification decision than classical NNs without feedback in the structure.

TABLE V.
PARAMETERS OF THE RNN NEURAL NETWORK

Parameter	Value	Parameter	Value
Solver	ADAM	MaxEpochs	30
Gradient Decay Factor	0.90	Mini Batch Size	200
Squared Gradient Decay Factor	0.99	Hidden Units# on L2	100
Initial Learn Rate	0.02	Hidden Units# on L3	100
Gradient Threshold	1	Hidden Units# on L4	6

In the general case of analysis with NNs, statistical features, like accuracy and loss, are evaluated. These features give additional properties of the data analyzed. In the paper we focus primarily on them. The training process of the RNN with the loss and accuracy values in time are represented in Fig. 15. Left and right vertical axes show the loss and the accuracy, respectively. The exponential trend of the loss is proved by a linear trend of the left-hand side axis having a logarithmic scale.

Fig. 15. Loss and accuracy of the training.

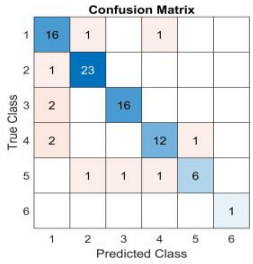


Fig. 16. Loss and accuracy of the training.

Note that the learning process of the RNN was relatively short, only 18.4 sec. We used the remaining 85 subtexts as test inputs of the RNN. It was established that the accuracy of the subtext classification is 87.06%, and the loss is 0.42%. Fig. 16 shows the confusion matrix of the 85 tested subtexts. This result seems very important because each fifth subtext can be identified by the corresponding Callimachus' class without a deeper interpretation of the subtext's meaning.

In Fig. 17 the dependence of the training time and test accuracy is represented vs. the number of subtexts, m ∈{5, 30}. The higher the number of subtexts is, the higher the learning time is. This duration remains under 20 seconds, proving the good performance of the evaluation.

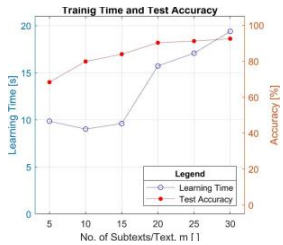


Fig. 17. Training time and test accuracy vs. number of subtexts.

The testing accuracy asymptotically increases with the number of subtexts, if $m \in \{5, 30\}$. Increasing the value of $m > 30$ destroys the testing accuracy. This effect is caused by the very low number of characteristic words (tokens) in the subtexts, removing the subtext from the global context of the original text.

D. *Impact of the Communication Errors on the Subtexts Classification*

Classical computer networks transfer text content in error-free mode. New transfer mechanisms offer quality-based services with nonzero but low error rates because of the significantly reduced price of these later transfers. We simulated the effect of transfer quality by introducing noise on the LSTM RNN performance during the classification of the subtexts. The noise was generated by replacing the original characters of the words in random positions with the next character in the alphabet. The last character, 'z' was replaced cyclically by the first character, 'a'. In most character replacements, the new words have no real meaning, so these words are assigned to the token category 'Other'. We should mention a few situations when the new word gets true meaning (i.e. 'war' has the third character 'r' replaced with the next letter 's', creating the word 'was'). It will probably influence the selection of the token category, too (see Table IV). The likelihood of these cases is very slight. Therefore, we did not consider them explicitly. According to Table II, this pollution of the original texts was executed before the text processing. The effect of this noisy modification was evaluated in the function of character error rate (CER), p, and the number of subtexts, m. For a relatively high character error rate ($p = 16\%$), the correlation between subtexts remains over 78% for a split ratio of the texts in $m = 20$ subtexts (see Fig. 18).

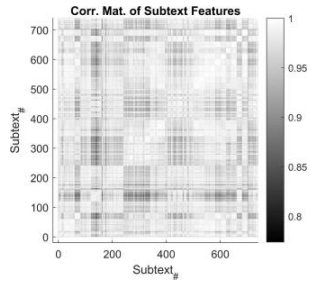
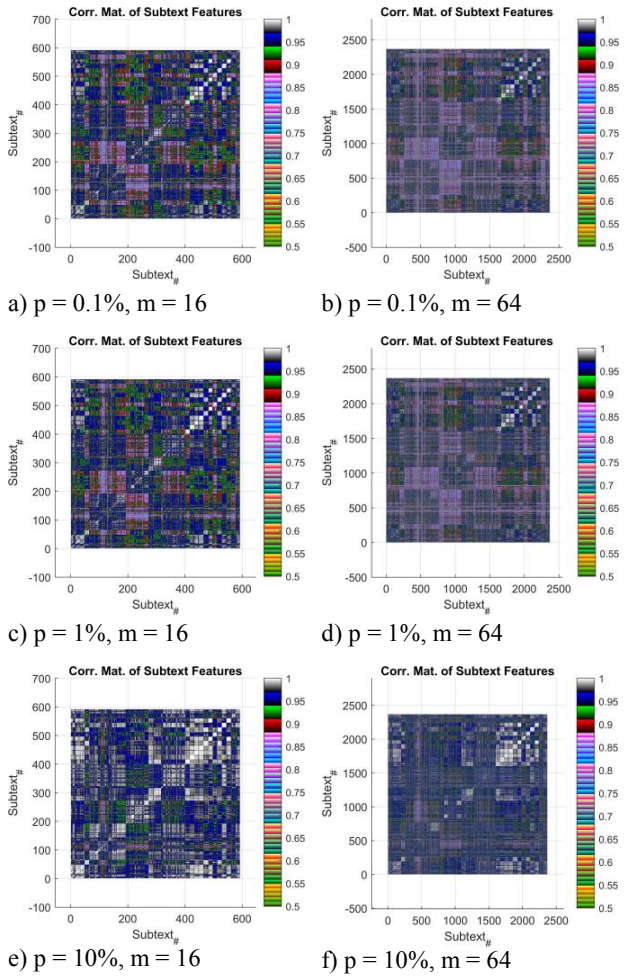


Fig. 18. Correlation matrix of subtext features (p = 16%, m = 20).

Based on Fig. 12, 13, and 18, we found that the minimum value (0.78%) of the correlation ranges is given by case $m = 20$. It is caused by the reduction of tokens in the subtexts corresponding to the error rate of the characters with probability, $p = 16\%$. In this sense, the effect of the pollution is reduced.

Fig. 19 illustrates the correlation matrixes between subtexts of different split ratios and error rate cases. The left (a, c, e) and right (b, d, f) hand side figures belong to the split ratio $m = 16$, and $m = 64$, respectively. Figure rows show character error rates of 0.1%, 1%, and 10%. The patterns of the two columns are very different because the number of subtexts is distinct. This implies considerable differences regarding the number of collected tokens from the subtexts, causing incomparable patterns of the corresponding correlation matrixes.

We realized that for low error rates, $p \in \{0.1\%, 1\%\}$ the patterns for $m = 16$ are very similar but not identical. The same property exists in the cases of $m = 64$. When the character error rate is large, $p = 10\%$, the pattern of the correlation matrix is significantly distinguishable from the cases of the same split ratio (see pairs c-e and d-f of Fig. 19, respectively).



a) p = 0.1%, m = 16 b) p = 0.1%, m = 64

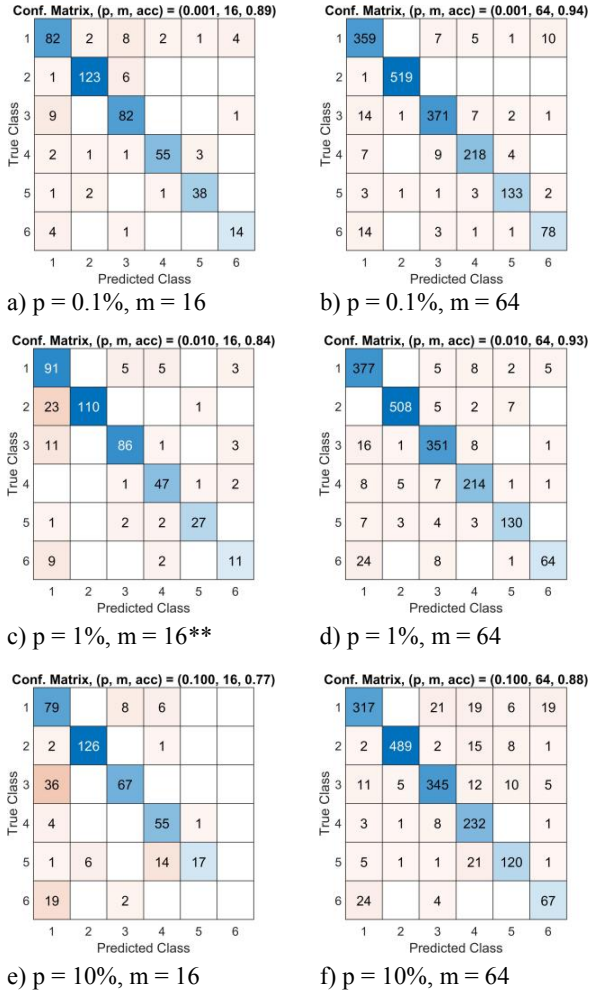
c) p = 1%, m = 16 d) p = 1%, m = 64

e) p = 10%, m = 16 f) p = 10%, m = 64

Fig. 19. Dependence of the correlation matrixes on the character error rate and no. of subtexts.

The white color of the pixels of the figures reflects a high correlation, being close to value 1. Pattern e of Fig. 19 stands for the same number of subtexts as patterns a and c. However, because it has a larger character error rate, the size of the subtexts is smaller, resulting in a smaller number of representative tokens enrolled into categories of Table IV. These remaining tokens correlate more to the common context, implying white-colored clusters with high cohesion of the neighboring subtexts.

The number of six Callimachus' classes remains unchanged, but the character error rate influences the accuracy of the subtext classification made with the LSTM neural network. The higher the value of the parameter p is, the lower the accuracy of the RNN classification of the subtexts is (see confusion matrixes a-c-e and b-d-f of Fig. 20, respectively). The higher the number of subtexts is, the higher the accuracy of the RNN classification becomes. This rule cannot remain valid for any parameter m value, because of the limited number of tokens in the subtexts. Other situations may occur when we have only one token in each subtext. The majority of the tokens alone have no context (i.e., 'war', 'god', 'king', etc.); just a few of them may be considered to have slight context (i.e., 'Zeus', 'Agamemnon', 'Prometheus'). This effect needs further study.



a) p = 0.1%, m = 16 b) p = 0.1%, m = 64

c) p = 1%, m = 16** d) p = 1%, m = 64

e) p = 10%, m = 16 f) p = 10%, m = 64

Fig. 20. Dependence of the confusion matrixes on the character error rate and no. of subtexts.

As the character error rate rises beyond 10%, the accuracy of the subtext classification based on LSTM RNN starts to decrease. Fig. 21 shows the dependence of the classification accuracy of the neural network on the character error rate in the case when $m = 50$.

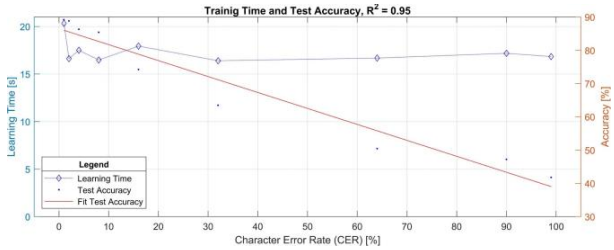


Fig. 21. Training time and test accuracy vs. character error rate (m = 50).

We determined that the accuracy of the ancient Greek text classification executed with the LSTM neural network when the texts are split into $m = 50$ subtexts is approximately linear dependent on the character error rate. The explicit equation is the following:

$$Accuracy\ [\%] \cong 0.9 - 0.5 \cdot p \qquad (6)$$

This equation has a minimum value of 40% when the character error rate approaches value 1. It means that if we strongly pollute these texts, the classification accuracy remains considerable. This property is typical of the ancient Greek texts. It remains an open question to address what is the case with the other textual data depending on their subject.

IV. Conclusions

In this paper we proved that the ancient Greek texts translated into archaic English language have an essential feature: extracting statistical properties of the most frequent 1000 grammatical tokens from the subtexts of 37 texts makes it possible to identify Callimachus' class of subtexts with the probability of 87.06%. The identification of the subtexts' class was accomplished by a Long-Short Term Memory type recurrent neural network. This result is worthy of note for two reasons: the meaning of the verbal content is not required to be considered in the classification process of texts; the learning time of the recurrent neural network takes less than one minute on an ordinary desktop computer. Because of the relatively small number of ancient texts accessible in English, the question of the usage of this proposed methodology can be addressed in detail in another research paper, and the methodology can be refined, if necessary. A possible continuation of this research work is the determination of the plot dynamicity of the texts and the classification of the texts based on their dynamicity. We plan to represent this new dimension of the texts in a multimodal way (e.g., image and sound). Long-Short Term Memory type of neural networks can classify subtexts of ancient Greek literary works even in the case of missing words. This method can be applied to help interpret the old manuscripts affected by the physical deterioration of their material. A possible continuation of the research work may be finding the appropriate methods and features to differentiate between lyric and prose form translations of the same text.

Acknowledgment

This work has been supported by the QoS-HPC-IoT Laboratory and project TKP2021-NKTA of the University of Debrecen, Hungary. Project no. TKP2021-NKTA-34 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the TKP2021-NKTA funding scheme.

References

[1] H. F. Hsieh, and S. E. Shannon, "Three approaches to qualitative content analysis," *Qualitative health research* vol. 15, no. 9, pp. 1277–1288, 2005.

[2] K. Krippendorff, Content analysis: An introduction to its methodology. California: Sage, 2018.

[3] K. A. Neuendorf, The content analysis guidebook. 2nd ed. London: Sage, 2016

[4] J. Kasperiuniene, M. Briediene, and V. Zydziunaite, "Automatic content analysis of social media short texts: Scoping review of methods and tools," *Computer Supported Qualitative Research. WCQR 2019. Advances in Intelligent Systems and Computing*, vol. 1068, A. P. Costa, L. P. Reis, A. Moreira, Eds. Cham, Switzerland: Springer, cop. 2020, pp. 89–101.

[5] J. Singh, G. Singh, R. Singh, P. Singh, "Optimizing accuracy of sentiment analysis using deep learning based classification technique," *Data Science and Analytics. REDSET 2017. Communications in Computer and Information Science*, vol. 799, B. Panda, S. Sharma, N. Roy, Eds. Singapure: Springer, 2018, pp. 516–532.

[6] J. R. Macnamara, "Media content analysis: Its uses, benefits and best practice methodology," *Asia Pacific Public Relations Journal*, vol. 6, no. 1, pp. 1–34, 2005.

[7] Y. H. Chang, C. Y. Chang, and Y. H. Tseng, "Trends of science education research: An automatic content analysis," *Journal of Science Education and Technology* vol. 19(4), pp. 315–331, 2010.

[8] P. Baranyi and Á. Csapó, "Definition and synergies of Cognitive Infocommunications," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 67–83, 2012.

[9] P. Baranyi, A. Csapó, and Gy. Sallai, Cognitive Infocommunications (CogInfoCom). Springer International, 2015.

[10] Carl Vogel and Anna Esposito, "Interaction Analysis and Cognitive Infocommunications," *Infocommunications Journal*, vol. XII, no 1, March 2020, pp. 2–9. DOI: 10.36244/ICJ.2020.1.1

[11] I. Boda, M. Bényei, and E. Tóth, "New dimensions of an ancient Library: the Library of Alexandria," in *CogInfoCom 2013. Proc. of the 4th IEEE International Conference on Cognitive Infocommunications*, (Budapest, Hungary, December 2-5, 2013,) pp. 537–542.

[12] *MaxWhere VR Even more*. [Online]. (2023-04-28) Available: http://www.maxwhere.com/ .

[13] I. Boda and E. Tóth, "Text-based approach to second language learning in the virtual space focusing on Callimachus' life and works," in *P. Baranyi Ed. CogInfoCom 2019. Proc. of the 10th IEEE International Conference on Cognitive Infocommunications*, (Naples, Italy, October 23-25, 2019,) pp. 439–444.

[14] M. A. Korteby, Z. Gál, and P. Polgár, "Multi dimensional analysis of sensor communication processes," *Annales Mathematicae et Informaticae*, vol. 53, pp. 169–182, 2021.

[15] A. Aldabbas, Z. Gál, K. M. Ghori, M. Imran, M. Shoaib, "Deep Learning-Based Approach for Detecting Trajectory Modifications of Cassini- Huygens Spacecraft," *IEEE ACCESS*, vol. 9, pp. 39 111–39 125, 2021.

[16] *Kaggle Database: 1/3 Million Most Frequent English Words on the Web*. [Online]. (2023-04-28) Available: https://www.kaggle.com/rtatman/english-word-frequency

[17] C. J. van Rijsbergen, Information Retrieval. 2nd ed. London: Butterworths, 1979. [Online]. (2023-04-28) Available: http://www.dcs.gla.ac.uk/Keith/Preface.html

[18] Van Halteren, H., "Source language markers in EUROPARL translations," in *Proc. of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, pp. 937–944.
[19] Lynch, G. and Vogel, C., "The translator's visibility: Detecting translatorial fingerprints in contemporaneous parallel translations," *Computer Speech & Language*, vol. 52, pp. 79–104., 2018.
[20] Koppel, M. and Ordan, N., "Translationese and its dialects," in *Proc. of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 1318–1326.

Zoltan Gal is currently a full professor at the Faculty of Informatics, University of Debrecen, Hungary. He earned MSc in electrical engineering and computer science from the Technical University of Timisoara, Romania and PhD in informatics sciences from the University of Debrecen. His scientific interest is focused on distributed processing and communication systems, sensor technologies and services in the Internet of Things.
He was the CIO of his institute for 20 years and developed the university-level metropolitan area highs-speed data network and services with over 10k Internet nodes. He is Cisco Certified Network Professional lecturer since 1999 and taught over five hundred network professionals in the field. Starting in 2015 he is head of the Centre of High-Performance Computing at his university. He is an IEEE member since 1996 and published over one hundred twenty scientific conferences and journal papers: He supervises his own R&D&I project called QoS- HPC-IoT Laboratory.

Erzsébet Tóth is an assistant professor at the Faculty of Informatics, University of Debrecen, Hungary. She had a degree in English language and literature and library and information science in 1995. In 2008 she obtained her Ph.D degree in information science and technology. In the doctoral dissertation she investigated the evaluation of search engine performance. She is involved in a virtual library project that focuses on the presentation of digitized library content in three-dimensional space, and she studies the enhanced possibilities of the English language teaching and learning in virtual learning environment. Since 2022 she has been an IEEE member.