# ANOMALY DETECTION IN ARABIC TEXTS USING N-GRAMS AND SELF ORGANIZING MAPS

Abdulwahed Almarimi and Asmaa Salem

Department of Computer Science, Bani Waleed University, Bani Waleed, Libya

***ABSTRACT***

*Every written text in any language has one author or more authors (authors have their individual sublanguage). An analysis of text if authors are not known could be done using methods of data analysis, data mining, and structural analysis. In this paper, two methods are described for anomaly detections: n-grams method and a system of Self-Organizing Maps working on sequences built from a text. there are analyzed and compared results of usable methods for discrepancies detection based on character n-gram profiles (the set of character n-gram normalized frequencies of a text) for Arabic texts. Arabic texts were analyzed from many statistical characteristics point of view. We applied some heuristics for measurements of text parts dissimilarities. We evaluate some Arabic texts and show its parts they contain discrepancies and they need some following analysis for anomaly detection. The analysis depends on selected parameters prepared in experiments. The system is trained to input sequences after which it determines text parts with anomalies using a cumulative error and winner analysis in the networks. Both methods have been tested on Arabic texts and they have a perspective contribution to text analysis.*

***KEYWORDS***

*Anomaly Detection, N-gram of Words, N-gram of Symbols, Self Organizing Map*

## 1. INTRODUCTION

In text processing, many problems are solved connected to authorship of texts, for example authorship attribution, external plagiarism, internal plagiarism, authorship verification, text verification. The Authorship Attribution problem is formulated as a problem to identify the author of the given text from the group of potential candidate authors. Some interesting approaches to solving of the problem can be found in [1], [2], [3] and plagiarism [4], [5],[6] but in both problems there exist some groups of comparable authors and comparable texts. It means the results of analysis can be compared according to texts or authors. In our problem the author is known [7][8] and we analyze each text as one extra text. In the solution of the problem, we use Self-Organizing Maps (SOM) models of neural networks [9]. A good description of Self-Organizing Maps extensions for temporal structures can be found in [10], where some of the extensions are usable for sequences. SOM models of neural networks are applied to time series in [11] and it inspired us to apply the same in a text analysis.

This paper is written in the following structure: The second section describes the background and statistics of some analyzed Arabic texts. The third section describes our developed method, the algorithm of Character n-gram Profiles. The fourth section describes the second method which we use in analysis of texts. It is a system of Self Organizing Maps. In the conclusion, we formulate summary of results and the plan of the following research.

## 2. BASIC BACKGROUND AND TEXT STATISTICS

In the text verification we used Arabic texts from [12]. The statistics of 6 Arabic texts are shown in the Table I.

We will use the following symbols and definitions:

- $\Gamma$ - an finite alphabet of letters; $|\Gamma|$ is the number of letters in $\Gamma$; in our texts;
- $V$ - a finite vocabulary of words in the alphabet $\Gamma$, presented in the alphabetic order; $|V|$ - the numbers of words in the vocabulary $V$;
- $T$ - text document; a finite sequence of words $T$; $T = \langle w1,..., w_n \rangle; w \in V; N$ - the number of words in the text;
- $T = \langle t_1, t_2 ..., t_{|T|} \rangle; |T|$ – the number of symbols in the text $T$;
- ${}^n g$ – symbol for n-gram (build on symbols);
- ${}^n V$ – a finite vocabulary of n-grams, $|{}^n V|$ is the number of different n-grams in the vocabulary ${}^n V$;

Let *P(A)* and *P(B)* be the profiles of two texts A and B, respectively. We studied the performance of various distance measures that quantify the similarity between two character *n*-gram profiles in the framework of author identification experiments. The following dissimilarity measure has been found to be both accurate and robust when the two texts significantly differ in length.

$$d(A,B) = \sum_{{}^n g \in p(A)} \left[ \frac{2(f_A({}^n g) - f_B({}^n g))}{f_A({}^n g) + f_B({}^n g)} \right]^2 \quad (1)$$

where *fA (g)* and *fB (g)* is the frequency of occurrence (normalized over the text length) of the *n*-gram g in text A and text B, respectively *P(A)* the set of all n-gram in the part A.

If the numbers of occurrences of in the two parts A and B of document are known, a function on n-grams can be defined by:

$$k_{A,B}({}^n g) = \frac{\#o_B({}^n g)}{\#o_A({}^n g)}, \quad (2)$$

and the formula (1) can be modified as (3) using formula (2) as follows:

$$d(A,B) = \sum_{{}^n g \in p(A)} \left[ \frac{2(|{}^n B| - |{}^n A| * k_{A,B}({}^n g)}{|{}^n B| + |{}^n A| * k_{A,B}({}^n g)} \right]^2 \quad (3)$$

Table.1. Statistics of 10 Arabic Texts

| | Name of texts | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A14 |
| # words | 94197 | 48358 | 51938 | 31656 | 39340 | 36977 | 93668 | 40076 | 60503 | 2168 |
| # symbols | 395065 | 198019 | 247448 | 135573 | 152905 | 155301 | 375430 | 163212 | 258346 | 11409 |
| #diff. words | 14110 | 9061 | 25755 | 10098 | 7036 | 3492 | 16384 | 9391 | 20920 | 1108 |
| # words by length | | | | | | | | | | |
| 1 | 6 | 48 | 183 | 81 | 14 | 7 | 89 | 690 | 12 | 28 |
| 2 | 13217 | 7358 | 5365 | 4816 | 6188 | 6397 | 15396 | 7456 | 8079 | 312 |
| 3 | **23287** 24.72% | **12130** 25.08% | 9353* 18.00% | **7795** 24.62% | 9619* 24.45% | 7575* 20.48% | 23520* 25.10% | **8405** 20.97% | **12393** 21.48% | **529** 24.40% |
| 4 | 22426* 23.80% | 11653* 24.09% | **9779** 18.82% | 6324 * 19.97% | **11476** 29.17% | **8231** 22.25% | **25075** 26.77% | 7850 * 19.58% | 10592* 17.50% | 456* 18.07% |
| Arabic<br>Latin<br>Max frq. 3-grams | الم<br>alm<br>3027 | واا<br>waa<br>1797 | واا<br>waa<br>4242 | نال<br>nal<br>789 | قال<br>qal<br>2294 | الم<br>alm<br>971 | الل<br>all<br>3468 | الا<br>ala<br>1647 | الم<br>alm<br>1983 | الم<br>alm<br>57 |
| Arabic<br>Latin<br>Max frq. 4-grams | الله<br>allah<br>1479 | فيال<br>fiyal<br>525 | هوهو<br>huhu<br>797 | فيال<br>fiyal<br>346 | الله<br>allah<br>1986 | لبصر<br>lbsar<br>2230 | الله<br>allah<br>2958 | ثمال<br>thmal<br>1030 | فيال<br>fiyal<br>841 | الله<br>allah<br>27 |

## 3. CHARACTER N-GRAM PROFILES METHOD

The method is based on similarity/dissimilarity of the text parts and their occurrences of n-grams in comparison to the complete text. We modify the dissimilarity measure defined by (3) using (2) to normalized dissimilarity measure *nd* as follows :

$$nd(A,T) = \frac{1}{\left|{}^{n}A\right|} * \sum_{{}^{n}g \in p(A)} \left[ \frac{\left|{}^{n}T\right| - \left|{}^{n}A\right| * k^{n}{}_{A,T}({}^{n}g)}{\left|{}^{n}T\right| + \left|{}^{n}A\right| * k^{n}{}_{A,T}({}^{n}g)} \right]^{2} \qquad (4)$$

where *T* is the complete text, *P(A)* and  is the set of all n-grams in the text part A. The denominator $\left|{}^{n}A\right|$ ensures that the values of this dissimilarity function lie between 0 (highest similarity) and 1.

The complete set of parameter settings for the proposed method is given in Table 2.

Table.2. Parameter settings used in this study.

| Description | Symbol | value |
|---|---|---|
| Character *n*-gram length | Arabic | 4 |
| Sliding window length | w | 2000 |
| Sliding window moving | s | 100 |
| Threshold of plagiarism free criterion | | |
| Real window length threshold | t2 | 1500 |
| Sensitivity of plagiarism detection | a | 2 |

### 3.1.N-gram Profile and a Style Function

Let W be a sliding window moving through the document of length w (in letters) and a step s (in letters). The window represents a text part and will be moved every time to the right by s letters. The profile of the window W is defined by the value *nd(W;T)*. It is possible to define the style function of a text T, using profiles of the moving windows as follows:

$$sf(i,T) = nd(W_i,T), i = 1..\lceil |T|/s \rceil, \quad (5)$$

where $W_i$ is a window, $\lceil |T|/s \rceil$ is the total number of windows (it depends on a text length). If *w* > s the windows are overlapping. It means, a text part in each window of the text will be evaluated in a comparison to whole text. The size of the window and the distance of the moving of window should have some influence for the stability of the style function *nd*. The different results are illustrated in the next Figure. The figure shows sf function of Arabic text A4, for 4-grams, the length of the moving window was 2000 letters. In the above panel the moving step was 100 letters and in the down panel 500 letters. In the figure, a middle line is drawn representing the mean of all *sf* values along with two more lines representing the +/- standard deviation [13].

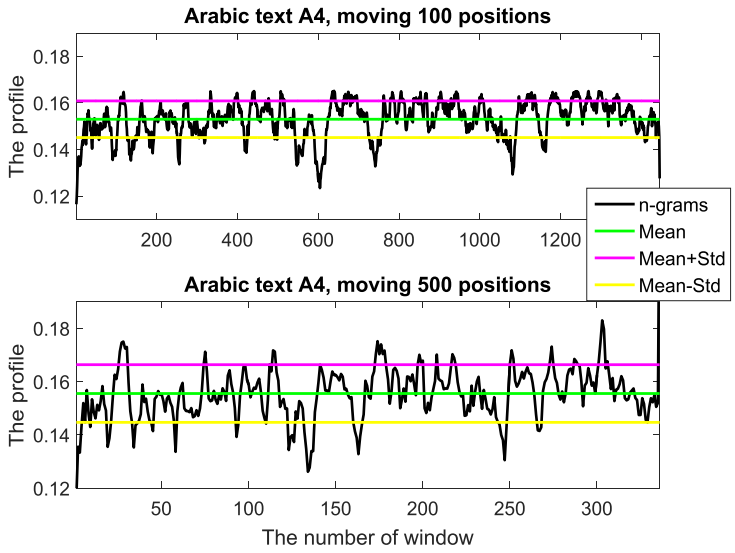


Figure.1. The style function of Arabic text A4, the window of the length 2000 letters moving by 100 positions using 4-grams (up) and moving by 500 positions (down).

### 3.2. Algorithm Covering Anomalies in Texts Parts

We expect that the style function is relatively stable (it does not change value dramatically) if the document is written by the same author. If the style function has very different values (some peaks [14]) for different windows, it is necessary to analyze the covered parts.

Let *M* be a mean value of the *sf* function values. The existence of peaks can be indicated by the standard deviation. Let *S* denote the standard deviation of the style function. If *S* is lower than a predefined threshold and profile values are less than *M* + *S*, then the text looks like consistent text of one author. The windows with the profile greater than *M* + *S* could be analyzed again.

**The steps of the algorithm:**

**Step 1:** We first remove from sf all the text windows with the profile less than $M + S$. The reduced text is $T'$. These windows correspond to unplagiarized sections probably.

**Step 2:** denote the style function after removal of the above described windows. Let $M'$ and $S'$ be the mean and the standard deviation of $sf(i', T')$

**Step 3:** The criterion (6) is defined to detect discrepancies. $$sf(i', W) > M' + a * S' \quad (6)$$

where parameter *a* determines the sensitivity of the discrepancies detection method. For the higher value a, the smaller number (and more likely problematic) sections are detected. The value of *a* was determined empirically at 2:0 [15] to attain a good combination of precision and recall. We used recommended value for the parameter *a*.

**Step 4:** Let #*dsf* be the number of windows which fulfil the condition (6). The percentage of discrepancies can be described by the formula (7)

$$P_{disc} = \frac{100 * \#dsf}{|W|} \quad (7)$$

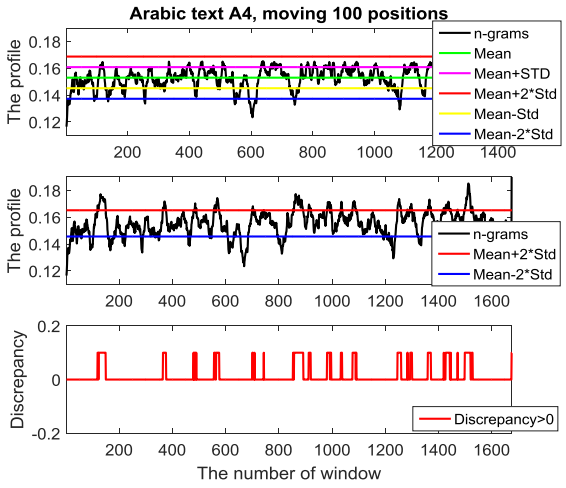


Figure. 2. The illustration of the algorithm on Arabic text A4, the window of the length 2000 letters moving by 100 positions. The binary function in the down panels indicates probably plagiarized passages (values greater than 0).

## 3.3. Evaluation of Character N-Gram Profiles Method

Our method covered the anomalies in such texts. Fig. 3 shows the application of 4-gram profile method on combined Arabic text (A4-A7). It means, the texts were created as an artificial combination of two parts of different texts. The style function of the combined text has different shape in the first part where discrepancies were identified. The percentage of anomalies is *Pdisc* = 25:75%. The percentage is higher than the supposed percentage (10%), meaning the text should have some anomaly.

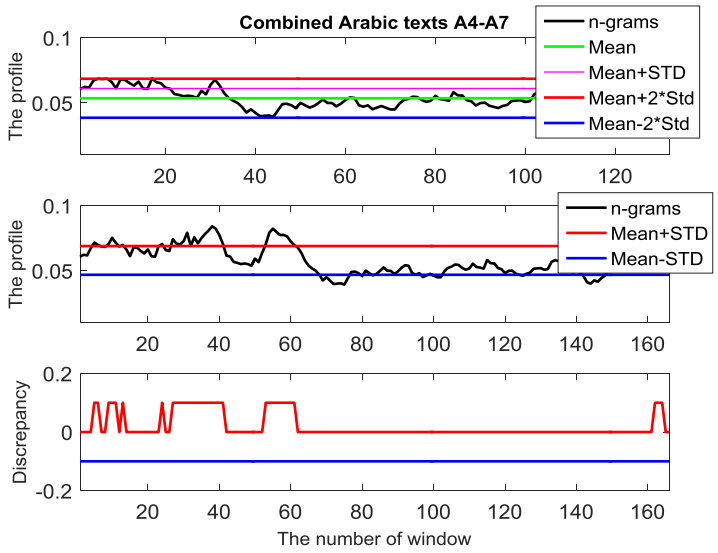


Figure. 3. The style function of combined two Arabic texts A4 (5347characters) and A7 (8155 first characters), the window moving by 100 positions using 4-grams. The binary function (the down panel) indicates problematic passages (high values). The percentage of discrepancies is Pdisc = 25:75%.

# 4. SYSTEM FOR ANOMALY DETECTIONS

## 4.1. Self Organizing Maps

The Self Organizing Map belongs to the class of unsupervised and competitive learning algorithms [9]. This type of neural network is used to map the n-dimensional space to the less dimensional space, usually two-dimension space. The neurons are arranged usually to the two dimensional lattice, frequently called a map. This mapping is topology saved and each neuron has its own n-dimensional weights vector to an input. If the input is represented by some sequence (for example, time series), when the order of values is important, then it is necessary to follow the order without changing it.

**The steps of the algorithm:**

1. **Initialization**: The weight vectors of each node (neuron) in the lattice are initialized to a small random value from the interval $\langle 0,1 \rangle$. The weight vectors are of the same dimensions as the input vectors.

2. **Winner identification for an input vector:** Calculate the distance of the input vector to the weight vector of each node. The node with the shortest distance is the winner. If there are more than one node with the same distance, then the winning node is chosen randomly from the nodes with the shortest distance. The winning node is called the Best Matching Unit (BMU). Let $i^*$ be index of the winning node.

3. **3. Neighbors calculation:** For this, the following equation is used:

$$h(i^*, i, t) = \exp\left(\frac{\left\| r_i(t) - r_{i^*}(t) \right\|}{\sigma^2(t)}\right) \quad (8)$$

Where $\sigma(t)$ means the radius of the neighborhood function, t is an iteration step, $r_i(t)$ and $r_{i*}(t)$ are the coordinates of units i and i_ in the output array.

4. **Weights adaptation:** Only the weights of the nodes within the neighborhood radius will be adapted. The equation for that is

$$W^{\rightarrow new} = W^{\rightarrow old} + \eta * h(i^*, i, t) * (\vec{x} - W^{\rightarrow old}) \quad (9)$$

where $W^{\rightarrow new}$ is the vector of the new weights, $W^{\rightarrow old}$ are old weights, $\eta \in$ (0,1) is the learning rate, $\vec{X}$ is the actual input vector. After the algorithm makes changes in the weights, it presents next random input vector from the remaining input vectors to input and continues with step 2 and so on until no input vector is left.

## 4.2. Description of the System Structure

In the first layer, it has $SOM_x$; x ∈ {words, w2-grams, w3-grams, s3-grams}, neural networks are trained to different sequences built according to the text D. The shape of the model is very similar to the model developed in [8], but here the different sequences are used for a training. The training of each $SOM_x$ is done on sequences $S_w$; $S_{w2g}$; $S_{w3g}$; $S_{s3g}$. The sequences are built according to the probability of n-grams [16].

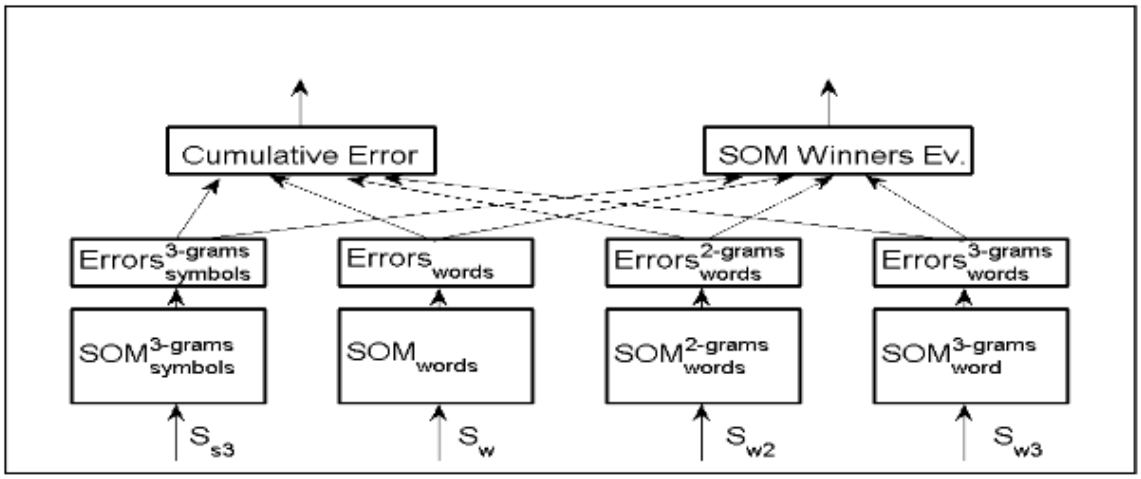


Figure.4. System for anomaly detections.

## 4.3. Description of the system computation

After the $SOM_x$ was trained it is possible to evaluate the quality of the training prepared by the evaluation of errors for all input vectors (all windows in the text). We will use a quantization error $Er_x$ defined by (10) as a measure of a proximity input vector $x^+$ to the learned winner vector $w_{i*}$ of $i^*$-th neuron (winner for input vector $x^+$) in the $SOM_x$

$$Er_x(x^+, w_{i*}) = \left\| x^+ - w_{i*} \right\|, \quad (10)$$

Using formula (10) it is possible to compute the vectors of quantization errors

$$\{Er_x(x^+(t), w_{i^*}(t))\}_{t=1}^{R} \qquad (11)$$

where R is the number of training vectors, t is the order of the member in input sequence. For the anomaly detections we will use thresholds developed by [11]. Let $\alpha$ be a significance level ($\alpha$ = 0:01 or $\alpha$ = 0:05). We suppose the percentage of normal values of the quantization error will be 100 * (1 - $\alpha$). Let $N\alpha$ be the real number such that a percentage 100 *(1 - $\alpha$) of the error values is less than or equal to $N\alpha$. Then

- Lower limit: $\lambda^- = N_{1\text{-}}\alpha_{/2}$
- Upper limit: $\lambda^+ = N\alpha_{/2}$

The important interval is $\langle \lambda^-, \lambda^+ \rangle$ the values out of it could be detected as anomalies.

The quantization vectors $Er_x$ and intervals $\langle \lambda^- s_x, \lambda^+ s_x \rangle$ are computed in the panels $Error_{sx,}$ x ; x ∈ {words, w2-grams, w3-grams, s3-grams} and they are used in two the following evaluations:

**1- Cumulative Error :**

$$CEr = \alpha_1 * Er_w + \alpha_2 * Er_{w2} + \alpha_3 * Er_{w3} + \alpha_4 * Er_{w4} \qquad (12)$$

where $\alpha_i$, i = 1; 2; 3; 4, $\sum_{i=1}^{4} \alpha_i = 1$ are parameters for a contribution of $Er_i$ to the cumulative error. The values of the parameters $\alpha_i$ should be chosen after the analysis of all errors. If the cumulative error has higher value as the threshold $h_{up}$ given by formula (13)

$$h_{up} = \alpha_1 * \lambda^+_{Sw} + \alpha_2 * \lambda^+_{Sw2} + \alpha_3 * \lambda^+_{Sw3} + \alpha_4 * \lambda^+_{Sw4} \qquad (13)$$

then the text needs some further analysis.

**2- Evaluation of SOM winners, clusters in SOM lattices:**

Fig. 5 shows the System of our method on combined Arabic text (A14). It means, the texts were created as an artificial combination of two parts from different texts (A1 and A4).

In the first part of Figure 5, we show the analysis of the cumulative error. The experiment was done with parameters $\alpha$ _1 = 0:3339; $\alpha$ _2 = 0:0314; $\alpha$ _3 = 0:0157; $\alpha$ _4 = 0:6189. The influence of the word probability in windows and 3-grams of symbols probability in the same window is higher than the others. The text needs some further analysis. It should have some anomaly. The second part shows SOM winners evaluations for windows moving through the text. The similar windows are grouped into clusters. We can follow more clusters than one. That means there should be some anomaly too in the text.

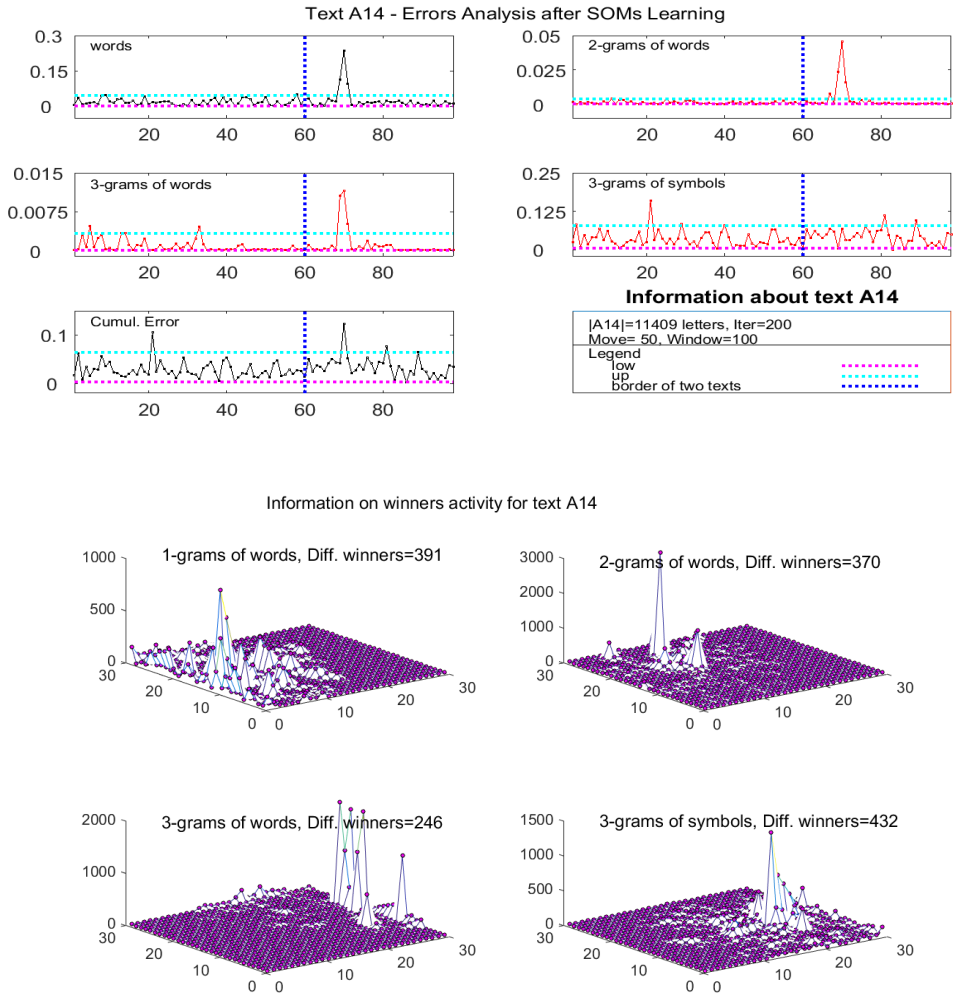


Figure .5. The evaluation of the cumulative error in the first part and evaluation of SOM winners in the second part for Arabic text A14. In the Figure 5, the first part shows the analysis of the cumulative error of the text A14. The experiment was done with parameters $\alpha$_1 = 0:3339; $\alpha$_2 = 0:0314; $\alpha$_3 = 0:0157; $\alpha$_4 = 0:6189: For all types of errors, there exist error values above the thresholds and for the threshold of the cumulative error too. The text should have some anomaly. The second part shows the clusters of SOM winners, more clusters illustrate that in the text should be some anomaly too.

## 5. CONCLUSIONS

In this paper, we developed two methods that compute some characteristics of texts. In both methods we illustrate results for Arabic text from the corpus [12],[3]. The first method is Character n-gram Profiles and the second method is a system for anomalies detections in a text. We illustrated results for Arabic texts. In this paper we showed the statistical analysis of Arabic texts and cover that using *4*-grams are better for it. The prepared analysis is very formal and in the next work we will try to apply some new accesses to the problem. Both methods are capable of covering anomalies of texts combined from two texts. The results from both methods trying to discover dissimilarities of text parts in each text show dissimilarities and they call for an attention to the text (or not) if the text parts were written by the same author (or not). We show the clusters of all four trained SOM networks. The training done on 1-grams of words, 2-grams of words, 3-gram of words

and 3-grams of symbols. According to the evaluation we can say that declared clusters of SOM networks trained for words better situation from symbols, they can give more information about clusters in some text.The clusters in the SOM lattice show similar characteristics of moved windows in the text. If it is possible to follow more clusters in the lattice, it is necessary to use some next analysis (the text probably has some anomalies). According obtained from these methods, the results for all texts from our point of view we can say that the second method System of Self Organizing Maps (SOM) is the most successful in the evaluation of the results, because we have many experiments of the text. Our next plan is to do many statistic tests to do evaluation and find better modifications of parameters.

## ACKNOWLEDGEMENTS

Thanks to Prof. Gabriela Andrejková, CSc. and Mgr. Peter Sedmák for their help.

## REFERENCES

[1] Neme, A. Pulido, J.R.G. Muño, A. Hernández, S. & Dey, T, (2015) "Stylistics analysis and authorship attribution algorithms based on self-organizing maps", Neurocomputing, 147 5 January 2015, pp 147–159.
[2] Stamatatos, E, (2010) "A survey of modern authorship attribution methods", J. Am. Soc. Inf. Sci. Technol. pp 538-556.
[3] Bensalem, I. Rosso, P & Chikhi, S, (2013) "A New Corpus for the Evaluation of Arabic Intrinsic Plagiarism Detection", CLEF 2013, LNCS 8138, pp 53-58.
[4] Eissen, S. M. Z. Stein, B & Kulig, M, (2006) "Plagiarism detection without reference collections", In: Decker, R., Lenz, H.J. (eds.) GfKl. Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin, pp 359-366.
[5] Stamatatos, E, (2006) "Ensemble-based author indentation using character n-grams", In Proceedings of the 3rd International Workshop on Text-based Information Retrieval 36, pp 41-46.
[6] Hassan, F. I. H & Chaurasia, M. A, (2012) "N-gram based text author verification", IACSIT press, Singapore 36, pp 67-71.
[7] Almarimi, A & Andrejková, G, (2015) "Text Anomalies Detection Using Histograms of Words", ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 5, No. 17, September 2015. ISSN: 2322-5157, pp 69-75.
[8] Almarimi, A. Andrejková, G & Sedmák, P, (2016) "Self Organizing Maps in Text Anomalies Detections", conference Cognition and Articial Life, Telč, 2016.
[9] Kohonen , T (2007) Self Organizing Maps. Prentice-Hall, 2 ed, 2007.
[10] Hammer, B. Micheli, A. Neubauer, N. Sperduti, A & Strickert, M, (2005) "Self-organizing maps for time series", WSOM 2005, Paris, pp. 1-8.
[11] Barreto, G.A & Aguayo, L, (2009) "Time series clustering for anomaly detection: Using competitive neural networks", Proceedings WSOM 2009 LNCS (5629), 28-36, 2009.
[12] King Saud University Corpus of Classical Arabic (2011). http://ksucorpus.ksu.edu.sa .
[13] Almarimi, A & Andrejková, G, (2015) "Document Verification using N-grams and Histograms of Words", IEEE 13th International Scientific Conference on Informatics, November 18-20, Poprad Slovakia, pp 21-26.
[14] Jurafsky , D & Martin, J.H (2000) Speech and Language Processing. Prentice-Hall, 1 ed., 2000.
[15] Stamatatos, E, (2009) "Intrinsic Plagiarism Detection Using Character n-gram Profiles", SEPLN Workshop on Uncovering Plagiarism, Authorship, and Social Software Misuse, PAN'09, pp 38-46.
[16] Stamatatos, E, (2010) "Authorship attribution based on feature set subspacing ensembles", International Journal on Artificial Intelligence Tools, 15.5, pp 823-838.

## AUTHORS

Master Degree from P. J. Šafárik University in Košice, Institute of Computer Science, Faculty of Science 2012, PhD. from P. J. Šafárik University in Košice, Institute of Computer Science, Faculty of Science2016, Faculty member at Bani Waleed University From 2017, Libya.

Master Degree from P. J. Šafárik University in Košice, Institute of Computer Science, Faculty of Science 2014, PhD. from P. J. Šafárik University in Košice, Institute of Computer Science, Faculty of Science 2019, Faculty member at Bani Waleed University From 2020, Libya.