Anomaly Searching in Text Sequences

Abdulwahed Almarimi, Gabriela Andrejková, Asmaa Salem
abdoalmarimi@gmail.com, gabriela.andrejkova@upjs.sk,
asmamostafa.salem@gmail.com

Pavol Jozef Šafárik University
Faculty of Science
Košice, Slovakia

Abstract

An analysis of some long text if authors are unknown or if it was written by one author is still an interesting problem and it could be done using methods of data analysis and data mining, and using structural analysis. In the paper, it is described a system of modified *S*elf-Organizing Maps working on probabilistic sequences built from a text. The sequences were built on letters and on words as n-grams, $1 \leq n \leq 4$. The system is trained to input sequences and after the training it determines text parts with anomalies (some different characteristics) using a cumulative error and a complex analysis. In tested long texts the system was successful, it covered a composition of texts.

1 Introduction

Each text is written in some genre, in some language and uses some grammar. It presents a sequence of letters, sequence of words, sentences, sections. We find some anomalies in the text, it means, we find text parts with some different characteristics based on n-grams in comparison to the full text or to the rest parts of the text. For example, anomalies should show if the text was composed from two or more parts written by different authors or if somebody manipulated with the text. The problem belongs to problems working with text and studying authorship attribution [Neme2015], [Stamatatos2010], and plagiarism [Eissen2006], [Stamatatos2006], and authorship verification [Hassan2012], but in both problems there exist some groups of comparable authors and comparable texts. It means that the results of analysis can be compared according to texts or according to authors. In our problem any author is known and we analyze each text as one extra text [Almarimi2015]. In the solution of the problem, we use Self-Organizing Maps (SOM) models of neural networks [Kohonen2007]. A very good description of Self-Organizing Maps extensions for temporal structures is in [Hammer2005], some of the extensions are usable for sequences. SOM models of neural networks were applied to time series in [Barreto2009] and they inspired for us to use it in a text analysis. In the text analysis we used English recommended texts from benchmark [CorE2011] and Arabic texts from [CorA2011]. In this contribution, we developed a new system for anomaly detections based on SOM model neural network and we applied it to Arabic and English long texts. The system covers anomalies in texts consisting of two texts written by different authors.

The paper has the following structure: The second section contains an information about used texts. In the third section, it is given a theoretical background for a construction of learning sequences (based on symbols

In: E. Vatai (ed.): Proceedings of the 11th Joint Conference on Mathematics and Computer Science, Eger, Hungary, 20th – 22nd of May, 2016, published at http://ceur-ws.org

and on words). In the fourth section, we describe our new developed system to anomaly detections. The fifth section contains an evaluation and illustration of results on English and Arabic texts. In the conclusion we give a resume of our work and some plan of further work in the area.

2 Sequences of Symbols, Sequences of Words

2.1 Theoretical Background for Sequence Construction

The applied method to an analysis of word or letter sequences is a probabilistic model. The working sequences will be presented by probabilities of their occurrences. It is possible to use individual relative frequencies of words or letters. But some of word combinations have a higher probability, some of them are used only once. It means, we will use a conditional probability of a word given by the previous words.

Notation:

- Γ - a finite alphabet of symbols; $|\Gamma|$ is the number of symbols in Γ; in our texts, Γ_A will be Arabic and Γ_E English alphabet;
- V - a finite vocabulary of words in a given text (the alphabet Γ) presented in the alphabetic order; $|V|$ - the numbers of different words in the vocabulary V;
- $d_1^N = \langle d_1, \ldots, d_N \rangle; d_i \in V$ - the text as a finite sequence of words; N - the number of words in the text d_1^N;
- $s_1^M = \langle s_1 s_2 \ldots s_M \rangle; s_i \in \Gamma$; - the text as a finite sequence of symbols; M – the number of symbols in the text s_1^M (including spaces);
- The probability of a complete sequence of words d_1^N can be represented by $P_d(d_1, d_2, \ldots, d_N)$ if each word is supposed as an independent event.
- The probability of a complete sequence of symbols s_1^N can be represented by $P_s(s_1, s_2, \ldots, s_M)$ if each symbol is supposed as an independent event.

According to [Jurafsky2000] the probabilities P_d and P_s can be decomposed using conditional probabilities in the following way:

$$P_d(d^N) = P(d_1)P(d_2|d_1)P(d_3|d_1^2)\ldots P(d_N|d_1^{N-1}) = \Pi_{i=1}^N P(d_i|d_1^{i-1}), \text{ where } d_1^{i-1} = \langle d_1 \ldots d_{i-1} \rangle \quad (1)$$

$$P_s(s^N) = P(s_1)P(s_2|s_1)P(s_3|s_1^2)\ldots P(s_N|s_1^{M-1}) = \Pi_{i=1}^M P(s_i|s_1^{i-1}), \text{ where } s_1^{i-1} = \langle s_1 \ldots s_{i-1} \rangle \quad (2)$$

The problem to compute the probabilities is solved using some simplification: an approximation of the probability of the word (symbol) given by all the previous words (symbols). The probability is depending on the probability of all preceding words but we will use only small number of them, for example $2, 3, 4$. It means, we work with $2, 3, 4$-grams.

The n-gram model approximates the probability of a word $d_i, 1 \leq i \leq N$ (symbol $s_i, 1 \leq i \leq M$) using all the previous probabilities of words $P_d(d_i|d_1^{i-n+1})$ (symbols $P_s(s_i|s_1^{i-n+1})$) by the conditional probability of preceding words $P_d(d_i|d_{i-1})$ (symbols $P_s(d_i|d_{i-1})$). It means,

$$P_d(d_i|d_1^{i-1}) \approx P_d(d_i|d_{i-n+1}^{i-1}), 1 \leq i \leq N, \qquad P_s(s_i|s_1^{i-1}) \approx P_s(s_i|s_{i-n+1}^{i-1}), 1 \leq i \leq M, \quad (3)$$

We have a formula for the general case of n-gram parameter estimation:

$$P_d(d_i|d_{i-n+1}^{i-1}) \approx \frac{C(d_{i-n+1}^{i-1} d_i)}{C(d_{i-n+1}^{i-1})}, 1 \leq i \leq N, \qquad P_s(s_i|s_{i-n+1}^{i-1}) \approx \frac{C(s_{i-n+1}^{i-1} s_i)}{C(s_{i-n+1}^{i-1})}, 1 \leq i \leq M, \quad (4)$$

where $C(d_{i-n+1}^{i-1} d_i)$ is the count of n-grams $d_{i-n+1}^i d_i$ and $C(d_{i-n+1}^{i-1})$ is the count of all $(n-1)$-grams d_{i-n+1}^{i-1}.

2.2 Sequences of symbols/words and n-gram of symbols/words

The text is a sequence of symbols in the alphabet Γ. We will map the given text to the sequence of occurrences computed from the text D.

1. The sequence S_s of symbol probabilities:
   - $C(s_i)$ – the number of occurrences s_i in the text D;
   - The sequence S_s is prepared by the formula:

$$f_s(s_i) = \frac{C(s_i)}{M}; 1 \leq i \leq M. \tag{5}$$

2. The sequence S_{sn} of n-gram of symbol probabilities:
   - The substring $s_i s_{i+1} s_{i+2} \dots s_{i+n-1}, i = 1, \dots M - n + 1$, $n = 1, 2, \dots$ is called n-gram of symbols; n - the length of the n-gram;
   - $C_n({}^n g)$ – *the number of occurrences* of n-gram ${}^n g$ in the text D;
   - $C_{n-1}({}^{(n-1)} g)$ – *the number of occurrences* of $(n-1)$-grams ${}^{(n-1)} g$ in the text D;
   - The sequence S_{sn} contains the frequencies of ${}^n g$ in the text D

$$f_{sn}({}^n g) = \frac{C_n({}^n g)}{C_{n-1}({}^{n-1} g)}. \tag{6}$$

3. The sequence S_w of word probabilities:
   - $C(d_i)$ – the number of occurrences d_i in the text D;
   - The sequence S_w is prepared using the following formula:

$$f_w(d_i) = \frac{C(d_i)}{N}; 1 \leq i \leq N. \tag{7}$$

4. The sequence S_{wn} of n-gram of word probabilities:
   - The subsequence $d_i d_{i+1} d_{i+2} \dots d_{i+n-1}, i = 1, \dots N - n + 1$, $n = 1, 2, \dots$ is called n-gram of words;
   - $C^w_n({}^n g)$ – *the number of occurrences* of n-gram ${}^n g$ in the text D;
   - $C^w_{n-1}({}^{(n-1)} g)$ – *the number of occurrences* of $(n-1)$-grams ${}^{(n-1)} g$ in the text D;
   - The sequence S_{wn} is prepared by formula:

$$f_{wn}({}^n g) = \frac{C^w_n({}^n g)}{C^w_{n-1}({}^{n-1} g)}. \tag{8}$$

3 System for Anomaly Detections

3.1 Self Organizing Maps

The Self Organizing Map belongs to the class of unsupervised and competitive learning algorithms [Kohonen2007]. This type of neural network is used to map the n-dimensional space to the lower-dimensional space, usually to two-dimensional space. The neurons are arranged usually to the two dimensional lattice, frequently called a map. This mapping is topology safe and each neuron has its own n-dimensional weights vector to an input. If 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 and do not change 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 among 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. **Neighbors calculation.** The following equation is used:

$$h(i^*, i, t) = exp\big(-\frac{||r_i(t) - r_{i^*}(t)||}{\sigma^2(t)}\big), \tag{9}$$

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 of the weights of the nodes within the neighborhood radius will be adapted using the equation:

$$\vec{w}^{new} = \vec{w}^{old} + \eta * h\left(i^*, i, t\right) * \left(\vec{x} - \vec{w}^{old}\right), \tag{10}$$

where $\vec{w}^{new}$ is the vector of the new weights, $\vec{w}^{old}$ are old weights, $\eta \in (0, 1)$ is the learning rate, $\vec{x}$ is the actual input vector.

After the algorithm make changes in the weights, it presents the next input vector from the remaining input vectors to input and continues with the step 2 and so on until there is no input vector left.

3.2 Description of the system structure

In the first layer, it has $SOM_x, x \in$ {words, w2-grams, w3-grams, s3-grams} neural networks they are trained to different sequences built according to the text T. The shape of the model is very similar to model developed in [Almarimi2016], but here the different analysed sequences are used for a training and an evaluation. The training of each SOM_x is done on sequences $S_w, S_{w2}, S_{w3}, S_{s3}$.

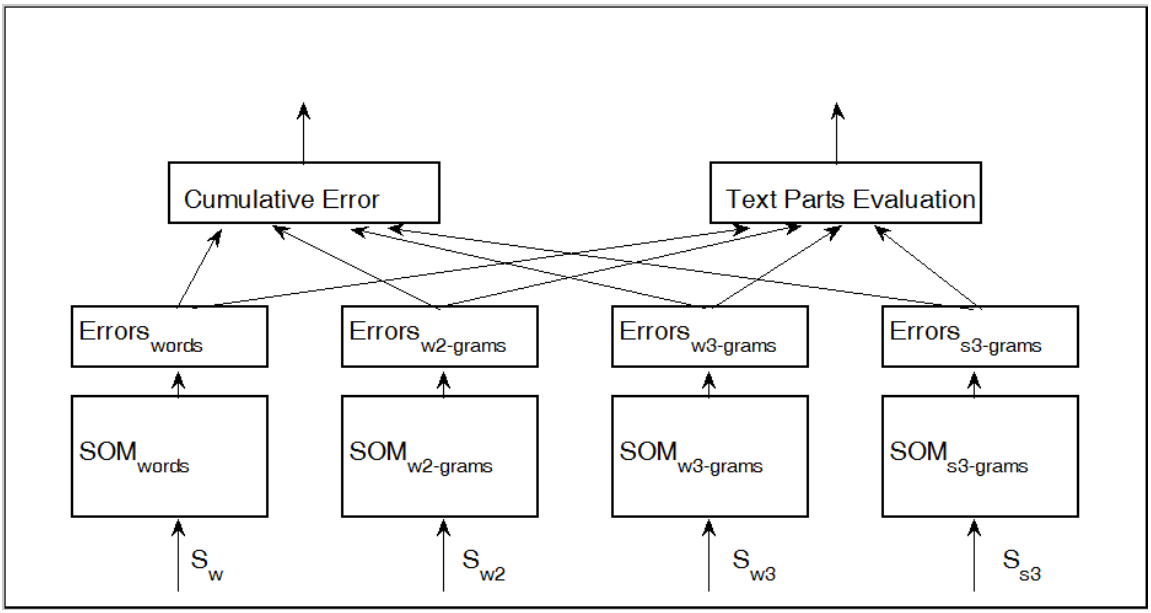

Figure 1: System for anomaly detections.The layer of SOM neural networks is trained to different sequences built to the same text. The layer of Errors blocks evaluates a quality of training. On the results of the quality is based a cumulative error and SOM winner evaluation.

3.3 Description of the system computation

After the SOM_x was trained it is possible to evaluate how good was the training prepared by an evaluation of errors for all input vectors (all windows in the text). We will use a quantization error Er_x defined by (11) as a

measure of proximity input vector $\mathbf{x}^+$ to the learned winner vector $\mathbf{w_{i^*}}$ of i^*-th neuron (winner for input vector $\mathbf{x}^+$) in the SOM_x.

$$Er_x(\mathbf{x}^+, \mathbf{w_{i^*}}) = ||\mathbf{x}^+ - \mathbf{w_{i^*}}||, \tag{11}$$

Using formula (11) it is possible to compute the vectors of quantization errors

$$\{Er_x(\mathbf{x}^+(t), \mathbf{w_{i^*}}(t)\}_{t=1}^R, \tag{12}$$

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 [Barreto2009]. Let α 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_α be the real number such that a percentage $100 * (1 - \alpha)$ of the error values is less than or equal to N_α. Then

- Lower limit: $\lambda^- = N_{1-\alpha/2}$
- 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 $Error_{S_x}, x \in$ {words, w2-grams, w3-grams, s3-grams} and they are used in two the following evaluations:

- Cumulative Error

$$CEr = \alpha_1 * Er_w + \alpha_2 * Er_{w2} + \alpha_3 * Er_{w3} + \alpha_4 * Er_{s3}, \tag{13}$$

  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 α_i should be chosen after the analysis of all errors. If the cumulative error has higher value than the threshold h_{up} given by formula (14)

$$h_{up} = \alpha_1 * \lambda^+_{S_w} + \alpha_2 * \lambda^+_{S_{w2}} + \alpha_3 * \lambda^+_{S_{w3}} + \alpha_4 * \lambda^+_{S_{s3}} \tag{14}$$

  then the text needs some next analysis.

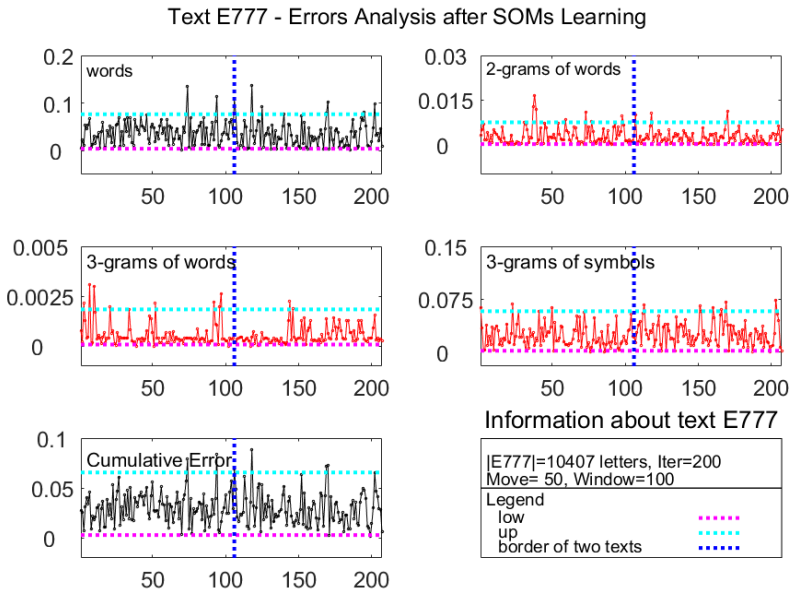

Figure 2: The layer of errors. Evaluation of the cumulative error for English text E777. The text E777 has 598 different words. The number of training vectors was 208, the number of iterations is 200.

- Evaluation of text parts

  The text T will be divided into $r, r > 1$ disjunctive parts, $T = T_1 T_2 \dots T_r$. For each text part $T_k, 1 \leq k \leq r$ the following evaluation will be done: $T_{-k} = T_1 \dots T_{k-1} T_{k+1} \dots T_r$ will be used as a training text and T_k will be a testing text. After the training using T_{-k} in our system, the testing text T_k will be evaluated using the quantization vector (12) and intervals $\langle \lambda^-, \lambda^+ \rangle$, the interval is built on training data. The percentage of values $x, x \in \langle \lambda^-, \lambda^+ \rangle$ express how the text T_k is similar to the training text. Evaluation of four sequences constructed from the texts gives more complex view to the text.

4 Evaluation

4.1 Data Preparation

In the text analysis we use English recommended texts from benchmark [CorE2011] and Arabic texts from [CorA2011]. In Table I we describe some information about 3 Arabic and 3 English texts, the texts A14 and E777 were constructed as a combination of two different texts (important for an illustration of our analysis). The letter position of the connection in both texts is shown in the thresholds of the analysis.

Table 1: Statistics of 3 English (E5, E14, E777) and 3 Arabic (A1, A4, A14) texts, the number of words by length for $1 - 10$ and maximal frequencies of 2 and $3 - grams$ for words and symbols.

	Name of Texts					
	A1	A4	A14	E5	E14	E777
Total-words	94197	31656	2168	93085	40721	1655
Total-symbols	395065	135573	11409	417899	216677	10407
# diff. words	14110	10098	1108	12929	7492	598
# words by length						
1	6	81	28	4018	1466	42
2	13217	4816	312	14663	5636	252*
3	**23287**	**7795**	**529**	**19875**	**9041**	**296**
	24.62%	24.28%	24.40%	21.35%	24.91%	17.88%
4	22426*	6324*	456*	18520*	7022*	228
	23.80%	19.97%	18.07%	19.89%	19.89%	
5	15887	5417	342	10109	4032	152
6	9159	3364	249	6927	3397	136
7	5559	2004	144	6461	2425	141
8	1978	802	41	3717	1695	122
9	921	349	29	2538	1119	107
10	336	182	9	1831	762	91
Max frq. sym	ال	ال	ال	he	he	th
Latin	al	al	al			
2-grams	22128	6545	474	11206	5337	218
Max frq. sym	الم	نال	الم	the	the	the
Latin	alm	nal	alm			
3-grams	3027	3027	57	6313	3276	127
Max frq. w	الله ، عليه	ألا ، ترى	عليه ، وسلم	[in the]	[of the]	[of the]
Latin	[allah,alyah]	[ala,tra]	[alyah,wasallm]			
2-grams	747	115	10	350	323	194
Max frq. w	صلى الله عليه	ألا ترى أن	صلى الله عليه	[it would be]	[one of the]	[the number of]
Latin	[salla,allah,alyah]	[ala,tra,ann]	[salla,allah,alyah]			
3-grams	745	40	10	37	168	10

4.2 Text E777, the cumulative error and text parts evaluation

Cumulative error. Figure 2 shows the analysis of the cumulative error. The experiment was done with parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ following the mean of each error $AveEr_j, j = 1, 2, 3, 4$. They were prepared according to the formula

$$\alpha_i = AveEr_i / \sum_{j=1}^{4} AveEr_j. \tag{15}$$

The used values were $\alpha_1 = 0.5614, \alpha_2 = 0.0440, \alpha_3 = 0.0087, \alpha_4 = 0.3859$. The influence of the word probability in windows and 3-grams of symbols probability in the same window is higher than the others. In the text E777 there exist some windows with the higher cumulative error than the threshold h_{up} defined by (14), the text needs some further analysis. It should have some anomalous features.

Text part evaluation. The text E777 was divided into 4 parts with the same lengths. Evaluation of their text parts is given in the Table 2. According to the similarity percentage in the columns "2-gram of words" and "3-grams of words" each part of text is not similar to the rest of the text E777 (the values are lower than 50%). The result confirms that the text is combined from two different texts.

Table 2: The percentage of a text parts similarity in English text E777. Each value present the similarity of testing part to training parts.

		Input Sequences			
Training Parts	Testing Part	words	2-grams of words	3-grams of words	3-grams of symbols
$\{T_2, T_3, T_4\}$	T_1	65.9574 %	48.9362 %	8.5106 %	51.0638 %
$\{T_1, T_3, T_4\}$	T_2	62.0000 %	46.0000 %	49.0000 %	78.0000 %
$\{T_1, T_2, T_4\}$	T_3	76.4706 %	29.4118 %	29.4118 %	74.5098 %
$\{T_1, T_2, T_3\}$	T_4	83.3333 %	35.1852 %	3.7037 %	61.1111 %

4.3 Text A14, the cumulative error and text parts evaluation

Cumulative error. Figure 3, the first part shows the analysis of the cumulative error of the text A14. The experiment was done with parameters $\alpha_1 = 0.3339, \alpha_2 = 0.0314.\alpha_3 = 0.0157, \alpha_4 = 0.6189$, computed according to the formula (15). 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 anomalous features.

Text part evaluation. The text A14 was divided into 4 parts with the same lengths. Evaluation of its text parts is given in the Table 3. The the similarity percentage in the columns "2-gram of words" and "3-grams of words" except the part T4 is lower than 50%, it means the text A14 is not consistent text. The result confirms that the text is combined from two different texts.

Table 3: The percentage of a text parts similarity in Arabic text A14.

		Input Sequences			
Training Parts	Testing Part	words	2-grams of words	3-grams of words	3-grams of symbols
$\{T_2, T_3, T_4\}$	T_1	54.7170 %	43.3962 %	43.3962 %	45.2830 %
$\{T_1, T_3, T_4\}$	T_2	77.9661 %	52.5424 %	32.2034 %	79.6610 %
$\{T_1, T_2, T_4\}$	T_3	70.3704 %	18.5185 %	22.2222 %	75.9259 %
$\{T_1, T_2, T_3\}$	T_4	54.7170 %	71.4286 %	67.8571 %	78.5714 %

4.4 Results of 40 Arabic and 40 English texts

The main goal of the system is to find some anomalies in the given long text. It means, our developed system have to be trained on some parts of the text and tested on the other parts. The system was evaluated for 40 Arabic texts [CorA2011] and 40 English texts [CorE2011]. In Table I we show some information about some of used texts.

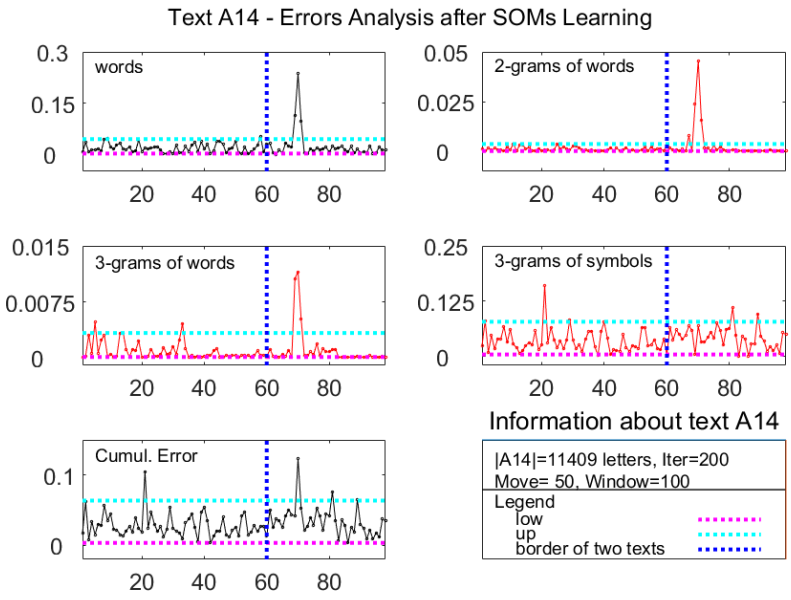

Figure 3: Evaluation of the cumulative error for Arabic text A14.

The strategy in the evaluation - to split each text into four parts, three of them were used for training and the fourth for testing. We had four possibilities for each text. The training and testing of each text was done for different input sequences to the system: words, 2-gram of words, 3-gram of words, 3-gram of symbols. The results of each text were analyzed according to criteria [Almarimi2016]:

- Good text – if the the values of percentage was higher than 75.000%.
- Critical text – if the the values of percentage was less than 75.000%. The critical text means that the tested part of the text gives under critical values.

Each text was evaluated 16 times, four times for each of the following methods: words, 2-gram of words, 3-gram of words, 3-gram of symbols. Results of the methods were combined into the last classification of each text (good or critical). In the Table 4, we show the information about 40 Arabic and 40 English texts.

Table 4: Information about Results for 40 Arabic and 40 English texts according to the used evaluation method

	Good		Critical	
Arabic Texts	{A1, A2, A3, A4, A5, A6, A8, A9, A10, A12, A13, A14, A16, A17, A18, A19, A20, A22, A23, A24, A26, A29, A33, A37, A40}	62.5 %	{A7, A11, A15, A21, A25, A27, A28, A30, A31, A32, A34, A35, A36, A38, A39}	37.5 %
English Texts	{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E19, E20, E21, E23, E24, E26, E29, E30, E32, E33, E34, E35, E36, E37, E38, E39, E40 }	80 %	{E16, E17, E18, E22, E25, E27, E28, E31}	20 %

5 Conclusion

In the paper we developed the system for the detection of anomaly parts in some text. The system was tested on 40 English and 40 Arabic texts and it is capable to cover the text with anomaly if the text is a composition of two texts. The results for English texts are better than for Arabic texts. Still it is necessary to do more statistic tests to the evaluation of the developed system and to find a better settings of parameters for Arabic texts mainly. It is possible to do it in experiments with parameters or to understand differences in the structure of both languages.

Acknowledgment

The research is supported by the Slovak Scientific Grant Agency VEGA, Grant No. 1/0142/15.
We thank to Bc. Peter Sedmák from Pavol Jozef Šafárik University in Košice for his help in a programming in Java.

References

[Almarimi2015] A. Almarimi and G. Andrejková. Document verification using n-grams and histograms of words. *IEEE 13th International Scientific Conference on Informatics* l (2322-5157), pp. 21-26, 2015.

[Almarimi2016] A. Almarimi, G. Andrejková and P. Sedmák. Self Organizing Maps in Text Anomalies Detections. *Proceedings of the conference Cognition and Artificial Life*, Telč, 2016.

[Almarimi2016] A. Almarimi: Dissimilarities Detections in Arabic and English Texts Using n-grams, Histograms and Self Organizing Maps. Pavol Jozef Śafarik University in Košice, PhD Thesis, 2016, 10, pp. 1-111.

[Barreto2009] G.A. Barreto and L. Aguayo. Time series clustering for anomaly detection: Using competitive neural networks. *Proceedings WSOM* 2009 LNCS(5629), 28-36, 2009.

[Bensalem2013] I. Bensalem, P. Rosso and S.Chikhi. A new corpus for the evaluation of Arabic intrinsic plagiarism detection. *CLEF* 2013, LNCS 8138 pp. 53-58, 2013.

[Chomsky2005] N. Chomsky. Three factors in language design. *Linguistic Inquiry* 36(1), 1-22, 2005.

[Durgin2005] N.A. Durgin and P. Zhang. Profile-based adaptive anomaly detection for network security. *SAND2005* (7293), 1-44, 2005.

[Eissen2006] S.M. Zu Eissen, B. Stein and M. Kulig. 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, 2006.

[Hassan2012] F. I. H. Hassan and M. A. Chaurasia. N-gram based text author verification. *IPCSIT*, IACSIT press, 36:67-71, Singapore, 2012.

[Jurafsky2000] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice-Hall, 1 ed., 2000.

[CorA2011] CorpusArabic. King saud university corpus of classical Arabic. http://ksucorpus.ksu.edu.sa, 2011.

[CorE2011] CorpusEnglish. pan-plagiarism-corpus-2011. `http://www.uniweimar.de/en/media/chairs/webis/corpora/pan-pc-11/`, 2011.

[Hammer2005] B. Hammer, A. Micheli, N. Neubauer, A. Sperduti, and M. Strickert. Self-organizing maps for time series. *WSOM* 2005, Paris, pp. 1-8, 2005.

[Kohonen2007] T. Kohonen. *Self Organizing Maps*. Prentice-Hall, 2 ed., 2007.

[Neme2015] A. Neme, J .R. Pulido, A. Muňoz, S. Hernández and T. Dey. Stylistics analysis and authorship attribution algorithms based on self-organizing maps. *Neurocomputing* pp. 147-159, 2015.

[Stamatatos2006] Stamatatos, E. Ensemble-based author identification using character n-grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval* 36, 41-46, 2006.

[Stamatatos2010] Stamatatos, E. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci. Technol* pp. 538-556, 2010.