TEXT CLUSTERING BY AUTHOR USING THE TIME SERIES MODEL

Liviu Sebastian MATEI[1], Stefan TRAUSAN-MATU[2]

In this article, we will introduce a novel model of clustering fragments of text by using the time series model. Given a set of book chapters, we want to cluster them by the author. The model that we are proposing makes use of the time series model in order to represent the text. Afterwards, we are going to use a distance algorithm called dynamic time warping for computing the distance between two chapters. In the end, using the distances computed a priori, we will evaluate different clustering algorithms in order to group the data.

Keywords: time series, dynamic time warping, text similarity, clustering

1. Introduction

Time series represent an ordered sequence of points collected at certain moments of time. An example of time series is the ordered sequence of n points: X={x_1, … , x_n} [1]. Each point corresponds to a certain moment of time. The sampling interval between points is usually constant, but this is not a mandatory requirement.

In this article we will present a novel method for clustering texts based on time series. In order to test and evaluate our model, we will pick chapters authored by Lev Nikolaevici Tolstoy and Feodor Dostoevsky and we will try to group them by the author. We picked Tolstoy and Dostoevsky for the evaluation of our model because they represent two well-known classical writers with two different writing styles: “Tolstoy's monologism with Dostoevsky's polyphony” [2]. The system doesn’t have any a priori knowledge regarding the texts, it just receives as input a set of chapters and it tries to cluster them by the author.

The method that we are proposing in this article will represent texts as a time series of cue phrases and punctuation signs and afterwards we will apply an algorithm in order to compute the distance between the two-time series. Further

[1] Eng., Dept.of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: liviumat@gmail.com

[2] Prof., Dept.of Computer Science, University POLITEHNICA of Bucharest; Senior Researcher, Research Institute for Artificial Intelligence of the Romanian Academy; Full member of the Academy of Romanian Scientists, Romania, e-mail: stefan.trausan@cs.pub.ro

on, based on these distances, we will apply a set of clustering algorithms in order to group the book chapters by author.

2. Related work

There is a lot of research done in the area of time series as they are used intensively in various domains like signal processing [3] or economics [4]. In text analysis, there is some work done around the computation of similar texts by using different metrics and taking into consideration the semantic relations between words [5, 6]. In a previous paper [6], time series were used in natural language processing and information retrieval. There is described a method in which texts can be mapped to time series. This article will extend that research by defining a new mapping function, ϕ, based on punctuation signs and cue phrases. Time series can also be used in natural language processing for “hot bursty events detection in a text stream” [7]. In this case the time series is represented by a set of documents in which the authors want to find a subsequence of documents that match a set of features called bursty features. The technique introduced is called “feature-pivot clustering” [7] which consists of 3 phases: identifications of the bursty features, grouping of the bursty features and finally determine the hot periods.

An interesting mapping from a set of tweets to a time series is done by O’Connor et al (2010), in which 1 billion Twitter messages are being extracted, parsed and analyzed in order to construct a time series that models the evolution of people’s opinion on different topics [8]. An extension to this research is done by considering also the sentiment strength and by analyzing also the part of speech [9].

In regards to the clustering of time series, Liao [15] makes a comprehensive analysis of the available techniques. For example, Golay et. al [10] introduces a method based on fuzzy c-means in order to cluster fMRI (magnetic resonance imaging) [11] data, which is modeled as a time series. Other approaches use agglomerative hierarchical clustering [12] or K-Means [13].

3. Texts as a time series of cue phrases and punctuation signs

In this article, we will present a novel approach of representing text as a time series of cue phrases and punctuation signs. Now, given a text, we will define the time series associated with it, by considering cue phrases and punctuation signs.

The first thing that we need to do is to define a translation function $\phi : W \rightarrow V^m$, where W represents all the words from the vocabulary and V^m represents an m-dimensional vector space where each dimension stands for an attribute that is considered in the computation. As mentioned by Matei and Trausan-Matu [6], ϕ maps between the elements of the text and the elements of the time series – it takes as input a token (ϕ is applied to all the tokens from the text) and generates an m-dimensional vector of attributes where each attribute represents a dimension.

Now, we will consider a particular case in which m=1 – we are considering one attribute – and V=R, the set real numbers – basically, in this case, ϕ will map from token to a numeric value. First, a preprocessing step is performed, in which we remove from the input text all the tokens that are not punctuation signs or cue phrases. Afterwards, ϕ will assign different values to all the punctuation signs and cue phrases.

For the cue phrases, we considered the following types [14]:

1. Continuation signals – additional information will follow in the text (ex: 'and', 'with')
2. Change of direction signals – in this case different alternatives to the current ideas will follow in the text (ex: 'but', 'otherwise')
3. Sequence signals – are used to introduce an order between different elements (ex: 'but', 'though')
4. Time signals – are used for specifying the moment in time when an action is happening (ex: 'now', 'after a while')
5. Illustration signals – introduce an example in order to clarify something (ex: 'such as', 'similar to')
6. Emphasis signals – with these, the author tries to highlight something (ex: 'a key feature', 'remember that', 'should be noted')
7. Cause, Condition, or Result Signals – indicate a condition, a cause or the presence of a result signal. (ex: 'if', 'for', 'while', 'then')
8. Spatial sign – used in order to fix a certain location (ex: 'below', 'near')
9. Comparison - Contrast Signals – in this case, these signals are used in order to compare two elements (ex: 'or', 'less than')
10. Conclusion signals – introduce a conclusion (ex: 'finally', 'as a result')
11. Fuzz signals – bring more clarification or specify more details/information regarding an idea. (ex: 'almost')

Based on these, we consider the following definition for the mapping function ϕ:

$$\Phi(w)=\begin{cases} 2,\text{ for } ',' \\ 3,\text{ for } ';' \\ 4,\text{ for } ':' \\ 10,\text{ for } '.' \\ 15,\text{ for } '!' \\ 15,\text{ for } '?' \\ 20,\text{ for } ''' \\ 100,\text{ for a continue signal} \\ 150,\text{ for change of direction signals} \\ 200,\text{ for sequence signals} \\ 250,\text{ for time signals} \\ 300,\text{ for a signal that shows something} \\ 350,\text{ for emphasis signals} \\ 400,\text{ for cause, condition or result signals} \\ 450,\text{ for spatial signals} \\ 500,\text{ for comparison} \\ 550,\text{ for conclusion signals} \\ 600,\text{ for fuzz signals} \end{cases} \tag{1}$$

By using the ϕ function, we are encoding all the values for punctuation signs and cue phrases in different numbers. With the current approach that we are proposing, we are not taking into consideration the actual value that is associated with a punctuation sign or cue phrase – this value is used only to represent graphically the time series associated with a text. The difference between the values is not important, the only important thing is that the values are different (an exception is represented by '?' and '!' which are associated with the same value). The values for punctuation signs are smaller so that in the graphical representation we can see clearly the difference between the two time series. Also, we defined a bigger difference between the various types of cue phrases so that we can differentiate better between them.

Using the a priori defined translation function, we can compute the time series representation for our texts. For the evaluation, we are considering chapters from 3 novels authored by Tolstoy and 3 from the ones authored by Dostoievsky:

- From Tolstoy: chapters 1, 2, and 3 from 'Anna Karenina' and chapters 1, 2 and 3 from 'War and Peace'

- From Dostoievsky: chapters 2, 3, and 4 from ‘The Idiot’, chapters 3, 4 and 5 from ‘The Brothers Karamazov’ and chapters 4, 5, and 6 from ‘Crime and Punishment’

Below we are presenting a graphical representation of the time series, based on the encoding schema mentioned before – on the Ox we are representing the time, ‘t’, while on Oy represents the actual value of the time series at the moment ‘t’ noted as TimeSeries(t) . We will show them in groups of two (the left one will belong to a chapter from Tolstoy and the one from right to Dostoievsky) so that we can visualize in comparison the difference between them:

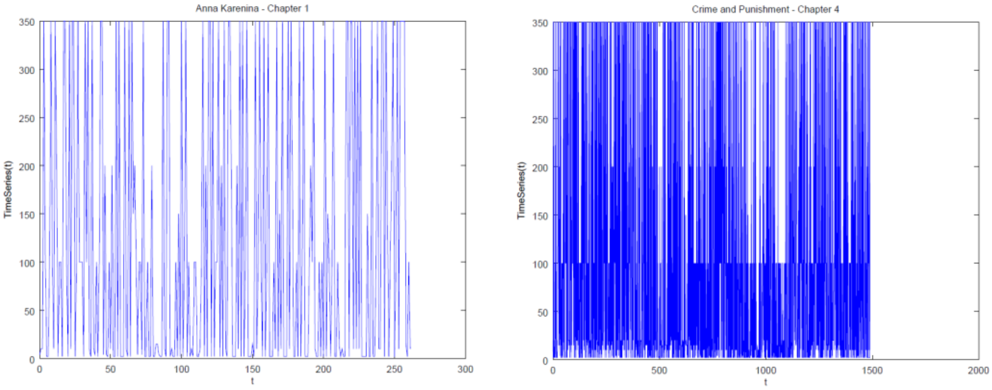

Fig. 1.‘Anna Karenina – Chapter 1’ in comparison with ‘Crime and Punishment – Chapter 4’

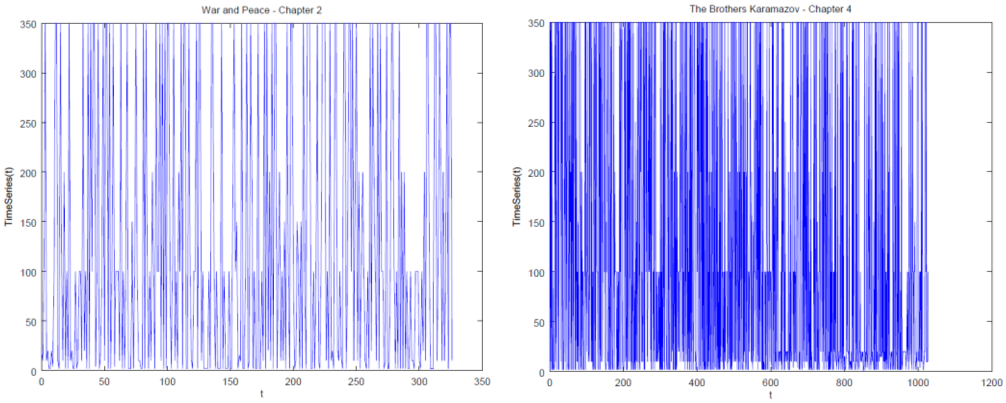

Fig. 2.‘War and Peace – Chapter 2’ in comparison with ‘The Brothers Karamazov – Chapter 4’

In both Fig. 1 and Fig. 2 the chapters that were authored by Tolstoy (‘Anna Karenina – Chapter 1’ and ‘War and Peace – Chapter 2’) look less ‘dense’ than the one authored by Dostoievsky (‘Crime and Punishment – Chapter 4’ and ‘The Brothers Karamazov – Chapter 4’). Also, we can notice that for the two chapters that were authored by Dostoievsky, we are having elements with higher values – these can be justified by the presence of more cue phrases inside the texts.

4. Distance computation between the time series of cue phrases and punctuation signs

Now, once we have defined ɸ, the next step is to define a distance function between the elements of the time series. Therefore, given two elements of the time series, e_1 and e_2, which in our case are numbers, we can define the following distance function:

$$dist(e_1, e_2) = \begin{cases} \text{MAX\_VALUE, if } e_1 != e_2 \\ 0, \text{if } e_1 = e_2 \text{ and } e_1 \geq 100 \\ 0.2, \text{if } e_1 = e_2 \text{ and } e_1 < 100 \end{cases} \quad (2)$$

We are defining the distance function to return the largest value possible from the domain – MAX_VALUE – in case the two elements of the time series do not match, 0 if the two elements match and they are cue phrases, and 0.2 for the case in which the two elements are punctuation signs. We choose this model of encoding in order to 'favor' more the cue phrases when doing the comparison than the punctuation signs – we consider cue phrases more important than punctuation signs for the overall similarity computation.

Given the previous defined ɸ and 'dist' functions we can now compute the distance between two-time series. The distance between the two is computed by using the dynamic time warping algorithm [15] – a dynamic programming algorithm used in order to compute the distance between two-time series that realigns the two subsequences in order to minimize the distance between them. Using this algorithm, we will compute the similarity between the chapters authored by Lev Tolstoy and Feodor Dostoevsky. We define the similarity of two texts t_1 and t_2 with the following formula:

$$\text{similarity}(t_1,t_2) = 1 - \text{DTW}(\text{ TimeSeries}(t_1),\text{TimeSeries}(t_2)\text{ }) \quad (3)$$

In formula (3) we consider that the function TimeSeries(t), where t is the input text, returns the time series associated with the text 't'. This time series is obtained after applying ɸ on each token of the text, as defined in (1). DTW is a function that computes the distance between two-time series, and it returns a numeric value in the interval [0,1], with 0 for perfect matches and 1 for time series that have nothing in common. As we are interested in the similarity between the two texts, we will subtract the distance from 1. Therefore, we can compute the similarity between all the pairs of chapters. The results that we obtained are listed in Table 1:

Table 1

Similarity between chapters authored by Tolstoy and Dostoevsky with DTW

	Anna Karenina - Chapter 1	Anna Karenina - Chapter 2	Anna Karenina - Chapter 3	War and Peace - Chapter 1	War and Peace - Chapter 2	War and Peace - Chapter 3	The Brothers Karamazov – Chapter 3	The Brothers Karamazov – Chapter 4	The Brothers Karamazov – Chapter 5	Crime and Punishment – Chapter 4	Crime and Punishment – Chapter 5	Crime and Punishment – Chapter 6	The Idiot – Chapter 2	The Idiot – Chapter 3	The Idiot – Chapter 4
Anna Karenina - Chapter 1	.90	.55	.57	.52	.57	.60	.53	.48	.50	.42	.45	.45	.46	.43	.45
Anna Karenina - Chapter 2	.55	.89	.54	.53	.54	.56	.52	.50	.51	.43	.46	.46	.46	.44	.46
Anna Karenina - Chapter 3	.57	.54	.90	.56	.57	.60	.58	.55	.57	.47	.52	.51	.51	.50	.51
War and Peace – Chapter 1	.52	.53	.56	.91	.55	.56	.55	.53	.55	.47	.51	.50	.51	.48	.51
War and Peace - Chapter 2	.57	.54	.57	.55	.91	.60	.55	.50	.53	.44	.48	.48	.48	.48	.45
War and Peace - Chapter 3	.60	.56	.60	.56	.60	.92	.55	.52	.54	.45	.50	.49	.49	.46	.48
The Brothers Karamazov – Chapter 3	.53	.52	.58	.55	.55	.55	.91	.56	.59	.50	.54	.54	.54	.53	.55
The Brothers Karamazov – Chapter 4	.48	.50	.55	.53	.50	.52	.56	0.9	.59	.52	.54	.55	.55	.55	.57
The Brothers Karamazov – Chapter 5	.50	.51	.57	.55	.53	.54	.59	.59	.92	.54	.57	.58	.57	.56	.59
Crime and Punishment – Chapter 4	.42	.43	.47	.47	.44	.45	.50	.52	.54	.89	.52	.53	.53	.53	.55
Crime and Punishment – Chapter 5	.45	.46	.52	.51	.48	.50	.54	.54	.57	.52	.90	.54	.59	.53	.55

Crime and Punishment – Chapter 6	.45	.46	.51	.50	.48	.49	.54	.55	.58	.53	.54	.91	.55	.55	.58
The Idiot – Chapter 2	.46	.46	.51	.51	.48	.49	.54	.55	.57	.53	.53	.55	.90	.54	.57
The Idiot – Chapter 3	.43	.44	.50	.48	.45	.46	.53	.55	.56	.53	.53	.55	.55	.89	.57
The Idiot – Chapter 4	.45	.46	.51	.51	.48	.48	.55	.57	.59	.55	.55	.58	.57	.57	.91

As we can see, all the values are in the interval [0, 1], the closer the score is to 1 the more similar the chapters are. It is important to note that with the model that we are proposing, even for the case of two identical texts, the score might not be 1 – for example in Table 1 we are not having any score of 1. This is due to the way we defined the distance function in (2). In order to obtain a similarity score of 1 we need to have a distance of 0 between the two-time series. However, if the time series contains punctuation signs, the distance between the two identical punctuation signs will be 0.2 according to (2) and as a result, the overall distance between the two-time series will be different than 0, leading to a similarity score different than 1.

In order to evaluate the results obtained, we compared them with a baseline by considering each chapter as a bag of words and applying the cosine distance. The vector representation of a chapter is formed from the tf-idf values associated with each word. The similarity results obtained for two of the chapters (Ana Karenina - Chapter 1 and The Brothers Karamazov - Chapter 3) are presented in Table 2:

Table 2

Similarity between chapters authored by Tolstoy and Dostoevsky with cosine distance

	Anna Karenina - Chapter 1	Anna Karenina - Chapter 2	Anna Karenina - Chapter 3	War and Peace - Chapter 1	War and Peace - Chapter 2	War and Peace - Chapter 3	The Brothers Karamazov – Chapter 3	The Brothers Karamazov – Chapter 4	The Brothers Karamazov – Chapter 5	Crime and Punishment – Chapter 4	Crime and Punishment – Chapter 5	Crime and Punishment – Chapter 6	The Idiot – Chapter 2	The Idiot – Chapter 3	The Idiot – Chapter 4
Anna Karenina - Chapter 1	1	.12	.11	.03	.02	.02	.04	.04	.03	.04	.04	.06	.05	.05	.05
The Brothers Karamazov – Chapter 3	.04	.03	.06	.04	.04	.03	1	.14	.1	.08	.05	.08	.09	.09	.09

As we can see, the greatest scores are indeed obtained with the chapters from the same book which was expected given that the number of words they have in common is greater. However, there is no correlation between chapters that belong to the same author - for example the similarity score between Anna Karenina Chapter 1 and Crime and Punishment Chapter 6 is 0.06, while the one between Anna Karenina Chapter 1 and War and Peace Chapter 2 is 0.02. This is expected given that although the novels are written by the same author, they have a different subject and therefore use different words.

5. Clustering texts with time series

Using the distances computed before, we can now try to cluster the time series associated with the novels. Therefore, multiple algorithms were tried out: K-Means [16], K-Medoids [17], density-based spatial clustering of applications with noise (DBSCAN) [18] and Hierarchical Agglomerative Clustering (HAC) [19].

In Table 1, the values obtained are close one with the others. Due to this reason, the results that we obtained with the DBSCAN algorithm and with HAC were not good. For DBSCAN the trend was to create one large cluster with all the chapters, due to its tendency to expand and acquire as many elements as possible. A similar thing happens with HAC, with the increase of the maximum allowed distance between clusters, in which case, eventually, we obtained one clusters with all the chapters, with a lower maximum distance the outcome was formed of multiple clusters, each of them having a few elements.

Another algorithm that was tested is K-Means. In this case the algorithm contains two parts: the distance computation between each element of the input and the centroid in order to assign each element of the input set to a cluster, and the second part, where we recomputed the new centroids of each cluster. By making use of the dynamic time warping algorithm, which computes the distance between two time series, we are able to make the correct assignation to the cluster. However, the problems appear with the second part of the algorithm, the computation of the new centroids. In this step, we need to compute an average of the existing elements so that we obtain the centroid. Thus, we tried two approaches:

- Using the Euclidian distance – however in this case there is a discrepancy in regard to the algorithm used, in order to compute the association of an element to a cluster – dynamic time warping – and the algorithm used to compute the new centroid, based on the Euclidian distance
- Using a generalized N dimension dynamic time warping algorithm – part of the research we generalized the DTW algorithm in order to

compute the distance between N time series. Besides the computation of the distance between the time series the algorithm also generates a summary of the time series, called warping path. As a result, we tried to approximate the centroid with the warping path. Unfortunately, the complexity of the algorithm becomes exponential in this case – O(N^p), where N represents the average number of elements the time series has and p the number of time series. Due to this, the algorithm doesn't scale, and in our case the time series have an average of 1000 elements making the algorithm not usable in terms of both execution time and memory consumption.

Therefore, the results that we obtained by using the K-Means algorithm were not satisfying – we obtained clusters with mixed results from both Tolstoy and Dostoevsky authors.

In order to overcome the problem that we mentioned before with K-Means, we considered a variation of the algorithm called K-Medoids. In this case, the only difference is that instead of computing an average centroid we pick one of the elements from the cluster that is found closest to all the other elements from the cluster. Thus, by running the algorithm on our input data set we obtained two clusters with the following data:

- Cluster 1: containing all the chapters that were authored by Dostoevsky: Crime and punishment - Chapter 4, Crime and punishment - Chapter 5, Crime and punishment - Chapter 6, The Brothers Karamazov - Chapter 4, The Brothers Karamazov - chapter 5, The idiot - Chapter 2, The idiot - Chapter 3, The idiot - Chapter 4, The Brothers Karamazov - Chapter 3
- Cluster 2: contains all the chapters that were authored by Tolstoy: Anna Karenina - Chapter 1, Anna Karenina - Chapter 2, Anna Karenina - Chapter 3, War and Peace - Chapter 1, War and Peace - Chapter 2, War and Peace - Chapter 3

In this case, all the chapters were grouped correctly based on the author. Therefore, we were able to cluster the chapters using a time series of punctuation signs and cue phrases.

In section 4, we defined a basic model based on the cosine distance and tf-idf in order to compare it with the time series model proposed in this article. Once we have the distances computed, we can use them in order to group the chapters into clusters with this technique as well. Thus, we applied the K-Medoids algorithm which grouped the chapters in two clusters:

- Cluster 1: Anna Karenina - Chapter 1, Anna Karenina - Chapter 2, Anna Karenina - Chapter 3
- Cluster 2: all the other remaining chapters

As we can see, using this technique the chapters were not correctly classified based on the author, which was expected given the similarity results.

6. Conclusions

In this article, we presented a novel approach for grouping book chapters based on the author by using a time series representation of punctuation signs and cue phrases. This representation combined with the K-Medoids clustering algorithm permitted us to cluster chapters that belong to novels authored by Lev Tolstoy and Feodor Dostoevsky. The model that we are proposing here can be easily extended in order to capture more details and particularities of the writers, for example the named entities or the part of speech of different words.

The similarity values that were obtained are in the range of [0, 1] – with the degree of similarity increasing as the value approaches 1. Although the actual scores that were obtained range from 0.42 – for the similarity between “Anna Karenina Chapter 1” and “Crime and Punishment – Chapter 4” and 0.6 as it is the case for the similarity between “Anna Karenina – Chapter 1” and “War and Peace – Chapter 3”, “Anna Karenina – Chapter 3” and “War and Peace – Chapter 3”, “War and Peace – Chapter 2” and “War and Peace – Chapter 3” – excluding the values that were obtained by comparing a chapter with itself. However, even though the values are placed in a small interval [0.42, 0.6] we can still determine the ones that hold the signature of Tolstoy and the ones that belong to Dostoevsky.

We also evaluated different clustering algorithms in order to determine which one integrates better with the dynamic time warping model and as we saw, the best results were obtained with the K-Medoids algorithm – 2 clusters, one containing the chapters from Tolstoy and the other one with chapters associated to Dostoevsky. Given the nature of the data, we obtain unsatisfactory results with the other clustering approaches which either create multiple clusters or have the tendency to generate one cluster with all the elements due to nature of the algorithm to expand – like it was the case for DBSCAN. As a result, the only algorithm that matched our expectation was K-Medoids and using it in combination with the time series representation and with dynamic time warping permitted to group the novel chapters by author.

In order to better assess this novel technique based on time series, we considered as a baseline a grouping of the chapters based on the cosine distance and tf-idf scoring. The results obtained with this standard technique were inferior to the ones generated by the time series method - with the combination of cosine distance and tf-idf we can at most group the chapters by novel but not author, which is our main goal and was achieved with the time series model introduced in this article.

REFERENCES

[1] *R. Weber*, "Time Series course notes". Retrieved January 2015, from University of Cambridge, Mathematics: http://www.statslab.cam.ac.uk/~rrw1/timeseries/t.pdf , 2007

[2] *A. Shukman*, "Bakhtin and Tolstoy", Studies in 20th & 21st Century Literature 9.1, Oxford, 1984

[3] *S. M. Kay*, "Statistical signal processing.", Estimation Theory 1, 1993

[4] *Z. Bar-Joseph, G. Gerber, D. K. Gifford, T. S. Jaakkola, and S. Itamar*, "A new approach to analyzing gene expression time series data", In Proceedings of the sixth annual international conference on Computational biology.1 pp. 39-48. ACM, 2002

[5] *X. Liu, Y. Zhou and R. Zheng*, "Sentence similarity based on dynamic time warping", Semantic Computing, 2007.

[6] *L. S. Matei, and S. Trausan-Matu*, "Document Semantic Distance based on the Time Series Model", RoEduNet Conference. Bucharest, 2016

[7] *G. P. C. Fung, J. X. Yu , P. S. Yu and H. Lu*, "Parameter free bursty events detection in text streams.", In Proceedings of the 31st international conference on Very large data bases, pp. 181-192, VLDB Endowment, 2005.

[8] *B. O'Connor, R. Balasubramanyan, B. R. Routledge and N. A. Smith*, "From tweets to polls: Linking text sentiment to public opinion time series", ICWSM, 11(122-129), 1-2, (2010)

[9] *P. Lai*, "Extracting Strong Sentiment Trends from Twitter", http://nlp.stanford.edu/courses/cs224n/2011/reports/patlai.pdf, 2010.

[10] *X. Golay, S. Kollias, G. Stoll, D. Meier , A. Valavanis and P. Boesiger*, "A new correlation based fuzzy logic clustering algorithm for FMRI", Magnetic Resonance in Medicine, 40(2), 249-260, (1998)

[11] *M. A. Lindquist*, "The statistical analysis of fMRI data", Statistical Science (2008): 439-464.

[12] *Y. Kakizawa, R.H. Shumway and N. Taniguchi*, "Discrimination and clustering for multivariate time series", J. Amer. Stat. Assoc. 93 (441) (1998) 328–340

[13] *T.W. Liao*, "Mining of vector time series by clustering", Working paper, 2005.

[14] *E. B. Fry, and J. E. Kress*. "The reading teacher's book of lists", Vol. 55., John Wiley & Sons, 2012.

[15] *T. W. Liao*, "Clustering of time series data - a survey", The Journal of the Pattern Recognition Society, pp. 1857-1874, 2005,

[16] *L. P.Stuart*, "Least Squares Quantization in PCM", IEEE Transactions On Information Theory, Vol. IT-28, No. 2, 1982

[17] *D. Arthur, S. Vassilvitskii*, "k-means++: The Advantages of Careful Seeding", Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. (1027-1035), Society for Industrial and Applied Mathematics, 2007

[18] *M. Ester, H. P. Kriegel, J. Sander, X. Xu*, "A density-based algorithm for discovering clusters in large spatial databases with noise", In Kdd, Vol. 96, No. 34 (226-231), 1996

[19] *C. D. Manning, P. Raghavan, H. Schütze*, "Introduction to Information Retrieval", Cambridge University Press, 2008