Identifying Cyclic Words with the Help of Google Books N-grams Corpus

Costin – Gabriel Chiru, Vladimir – Nicolae Dinu
Department of Computer Science and Engineering
Politehnica University from Bucharest
Bucharest, Romania
E-mail: costin.chiru@cs.pub.ro, vladimir.dinu92@yahoo.com

***Abstract*—In this paper, we present an application for identifying English words whose use is cyclic or regularly varies in time. The purpose of the developed application was to build a cross-platform system for indexing and analyzing the graphs of words usage over time. For words indexing, we used the data provided by the Google Books N-grams Corpus, which was afterwards filtered using the WordNet lexical database. For identifying the cyclic or regularly varying words, we used two different algorithms: autocorrelation and dynamic time warping. The results of the analysis can be visualized using a web interface. The application also offers the possibility to view the evolution of the use frequency of different words in time.**

Keywords-cyclicity detection; dynamic time warping; autocorrelation; Google Books N-grams Corpus; WordNet.

I. Introduction

This paper presents an application capable of indexing and analyzing the unigrams dataset from Google Books N-grams Corpus [1] for establishing which are the words that vary regularly in time, following a cyclic pattern. The analysis is done based on the graphs generated from the number of uses of each word in the publications from 1800 until 2008 that were indexed by Google. The results provided by the application are the words that were identified as being cyclic, along with the years where these words were cyclic and with the length of the cycle in years after which the pattern repreats.

The identification of cyclic words may prove useful in predicting many types of events, starting from the meaning of the cyclic word that was found. For example, if the word is a generic type of events, such as "rebellion", "revolution" or "war", they might suggest that such an event is about to happen. These results could also be used in the economic field, since if the word represents a salable product, it is possible that the public interest in that product has grown, and thus both sales and stocks of companies selling that product will possibly grow.

The paper continues with a short presentation of the researches developed starting from the Google Books Ngram Corpus. In Section 3, we present the details of our application, along with the data sources used, and afterwards we detail the two algorithms used for the identification of the cyclic words. Section 5 contains the obtained results and a comparison of the used algorithms from the perspective of these results. The paper ends with our final conclusions and with our plan to continue this research.

II. Similar Approaches

In [2], the authors analyzed the evolution of 107 words from English, Spanish and Hebrew over 208 years (from 1800 to 2008). The purpose of this research was to highlight the co-evolution of language and culture. The pronounced changes seen in the language dynamic during the conflict periods revealed the fact that the correlations that appear between words in time are influenced by co-evolutionary social, technological and political factors. The authors conclude that the birth of different words is most commonly related to new social and technological trends. Moreover, a new word requires some time to get into regular use and the authors established this period to be of 30-50 years.

Roth [3] examined the role of different autonomous function systems (such as the economy, science, art, religion, etc.) in 3 different societies (English, French and German) with the purpose of ranking these systems according to the public oppinion. He assumed that the public oppinion related to each such system may be expressed as the number of times words related to that system were used each year. In this sense, he used the graphs offered by Google Ngram Books Viewer to visualize the evolution in the use frequencies for the words related to the functional systems during 1800 and 2000. The results showed that even though everybody speaks about the economization, fact supported by the increase use of terms from the economic domain, the ascending tendencies of these terms stopped in each of the analyzed languages before reaching a dominant position. Moreover, for English, at the beginning, law was the dominant functional system, followed by religion and arts, while in the end policy was the main system, followed by law, health and education. For French, initially art ranked first, followed by religion, justice and policy. At the end of the analysis, on an ascending trend can be found policy, followed by art and economy. In the German case, the early 19th century was dominated by law as well as science, art, and religion. At the end of the analyzed period, policy was the main functional system, followed by legal system, art and science.

Acerbi et al. [4] analyzed the trend in using emotional words in the 20th century books using six lists of terms denoting fellings (such as anger, disgust, fear, happiness, sadness and surprise) that were previously used in a study on Twitter. The initial study revealed that changes in the frequency of use of different emotional words are caused by major events from human life, such as the death of a popular person, public tensions or natural disasters. Therefore, the authors hoped that, by using Google Books Ngram Corpus, they will be able to extend the proportion of the study.

The results of the study showed a descending trend in using emotional words in the last century, except for the last half, when, in american books can be seen an increase in the use of emotional words, compared to british books. Corroborated with this, the authors also investigated the difference between words and phrases related to the individual (e.g., independent, individual, unique, self, solitary, personal) and to the collective (e.g., team, collectively, set, group, union) showing that the former category has seen a great increase in american books during 1960 and 2008, while the others did not.

Another conclusion of the study is that there can be distinguished periods of happiness and sadness and that these periods are correlated with important historical events: the sadness peak corresponds to the WWII, while for happiness there are two peaks, one in 1920 and the other in 1960. Morerecently, it can be observed a sadness period starting in 1970 with an increase in hapiness in the last years of study.

Google Books Ngram Corpus was used in [5] to help compare different methods for estimating word relatedness. The authors used this corpus as common corpus for six corpus-based methods (Jaccard coefficient, Simpson Coefficient, Dice Coefficient, Pointwise Mutual Information, Normalized Google Distance and Relatedness based on Tri-grams). Afterwards, they compared the results obtained by the six methods on this corpus with the human ratings provided for some synonymy pairs. The comparison showed that the most accurate method is the Relatedness based on Tri-grams, which led to a Pearson correlation coefficient of 0.916. In the same time, the research proved the accuracy of the data from the Google Corpus, and stated the fact that all the analyzed metrics could be used on the n-grams offered by Google.

Another study, conducted by Wijaya and Yeniterzi [6] had the purpose to analyze the changes that occured in the meaning of a word over time. For that, they decided to analyze the evolution of the meaning of the words co-occurring with it over time. Thus, they used k-means clustering along with topic modelling over time, a topic modelling that has the advantage of using time as an explicit factor influencing the structure of a document.

As it can be seen from the above researches, time series analysis built on the data provided by Google Corpus has been used for various purposes, but none of them is similar to the approach proposed in this paper.

III. Implementation Details

A. *Used Resources*

For this research, we used data from two important sources: Google Books N-grams Corpus and WordNet.

1) *Google Books N-grams Corpus*

The data used in this project was extracted from the Google Books N-grams Corpus [1]. The corpus, created in 2009, contains the words written in over 5 million books published between 1500 and 2008. The corpus is made of over 500 billion words in 7 languages: English, French, German, Spanish, Hebrew, Russian and Chinese.

In this research, we only used the unigrams dataset from the corpus, this being formed with two types of files: the ones containing on each line information about the number of uses of a different word, and another one containing the total of words indexed for each year, that is used for normalization.

Although many researchers turned to this corpus for their experiments, there are also scientist criticizing it due to the errors generated by Optical Character Recognition (OCR) algorithms used to digitize content that was not digitally available. Another critique of this corpus is related to the insufficient data that it contains, only about 4% of all the books ever published being included in the corpus, thus lacking to offer a good coverage of all topics.

The corpus authors acknowledge these shortcomings and mention that the dataset is more relevant starting with 1800. Thus, we will restrict our analysis to the period 1800 -2008.

2) *WordNet*

The second data source for this research is represented by the WordNet lexical base [7] built at Princeton University. It contains only English words grouped based on their part-of-speech (nouns, verbs, adjectives and adverbs) and on their semantic, clustering words with similar meaning in synsets.

B. *Modularization*

The architecture of this application follows a three-tiers organization, with the data access module on the first tier, the services modules for offering the functionalities on the second (the logic tier) and the presentation tier, viewed by the user, which contains three modules: the indexer, the analyzer and the graphical user interface (the GUI of the application).

1) *Data Access Module*

This module contains the access logic to the tables from the database. The database contains a table “total”, which saves the data referring to the entire unigram corpus used in this research; 26 tables (one for each letter) containing the words starting with that letter; and another 26 tables, similar to the 26 before, that capture the results obtained by our application for each word from the databse.

The “total” database contains the aggregated information about the corpus, each of its entry being specific to a different year and containing the total number of words, pages and volumes that were indexed by Google for that year. This table is used for normalizing the data from the other tables.

Each of the 26 tables containing the words have information referring to the number of appearances, number of pages and volumes in which the words starting with the letter specific to the table were found.

Finally, the 26 results tables contain the data about each words’ cyclicity obtained using the algorithms described in the next section.

The choice of saving the words in 26 tables was done for efficiency reasons, since the application was very slow when the words were saved in a single table.

2) *Services Module*

This module contains the services needed for running the application (the business logic). This module contains two types of resources: on one hand, there are the services for accessing the database from the previous tier, and on the other there are implementations for the two algorithms used for identiying the words' cyclicity.

The services from the first class are responsible for the create, read, update, and delete (CRUD) operations with the "total" table, data selection for a given word, memorizing the generated graphics in the database, adding the analysis results for a given word using the two algorithms, the selection of the best results obtained, etc. Also, for all these operations is used a caching system for improving the application performances.

The second class of resources contains the implementation of the autocorrelation and the DynamicTimeWarping (DTW) algorithms. These resources receive as input the normalized graph of a given word and will output the data obtained after running each algorithm.

3) Indexer Module

This module deals with the indexing of the data from the n-grams files created by Google and, since it is responsible for the data acquisition, is the first module to be. Thus, it initially parses the files for the words starting with each letter and saves them in one of the 26 tables, and then it indexes the aggregation file for filling the "total" table.

The indexer module is also responsible for filtering the data, with the help of the WordNet lexical base, considering several criteria: the word's length should be greater than 2; the characters composing the word can only be letters, quotes, or dashes; the word cannot contain more than 3 identical consecutive characters; the word should be also present in the WordNet lexical base; information about the word's use should be present for at least 10 different years; and the dataset should contain information for at least 95% of the years in which the word was used.

The filtering's purpose is to ensure the accuracy and representativeness of the analyzed data, and thus, the words that did not respect at least one of the above criteria was removed from analysis.

The number of words that were kept for further analysis (after filtering) is presented in Table 1, along with the obtained results derived from their analysis.

4) Analyzer Module

This module is the main part of the application, being responsible for identifying the words' cyclicity. Thus, it runs the algorithms presented in the next section on the data that was saved in the database by the indexer module. It outputs the best results obtained by each of the algorithms.

First, the module iteratively selects the raw data of each word's usage. Then, this data is normalized, on a yearly base, with the help of the aggregated data from the "total" table, specifying the total number of all words' usage for each indexed year. Thus, the count of words' usage is transformed in the frequency of words' usage, having values in the [0, 1] interval. Finally, the algorithms for detecting words' cyclicity are run on this data.

To obtain the best possible results, when applying the algorithms, we varied the running parameters, which led to multiple runs of the algorithms for the same word. The algorithms have two parameters that can be adjusted: the length of the interval in which the words' cyclicity is tested and the length of the cycle.

Regarding the first parameter, for both algorithms, we varied the starting date of the interval, with a rate of 10 years, starting from 1800 and ending with 1980. Thus, for each word, we tried to detect cycles in the intervals: [1800, 2008], [1810, 2008], ... [1980, 2008].

For the second parameter, we varied the length of the cycle that we were looking for depending on the length of the interval in which words' cyclicity was tested. Thus, we varied the length of the searched cycle starting from one sixth of the whole interval and ranging up to one third of the interval. For example, for a word graph in the [1948, 2008] interval, the length of the cycle may vary between 10 ((2008-1948)/6) and 20 ((2008-1948)/3) years.

After running the two algorithms for detecting the words' cyclicity with all the presented choices for the two parameters, the best obtained results, along with the setting of the parameters that led to these results (the starting year of the analysis and the length of the cycle), were saved in the results table from the database. In Table 1, we present the best results obtained.

5) Graphical User Interface Module

The last module of the application is responsible for accessing and presenting the results of the analysis. This module is a web interface that is presented in Figure 1. It allows the selection of a given word and the visualization of its usage in time, along with the best results obtained using the two algorithms. It also shows the aggregated data, presenting general information about the dimension of the data indexed by Google and of the data retained by our application.

IV. USED ALGORITHMS

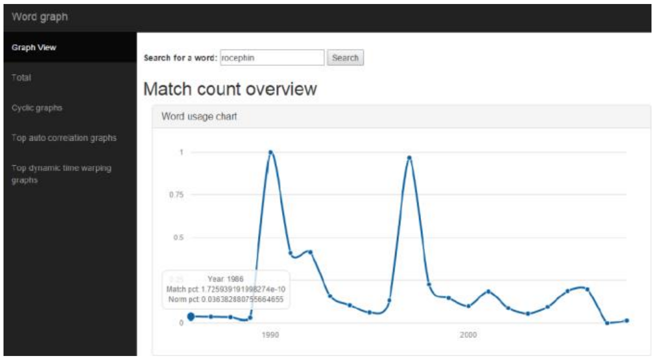
In this section, we will present the two algorithms that we used for detecting the words' cyclicity: autocorrelation and DTW.

A. Autocorrelation

Autocorrelation [8] is an analysis method for time series used for determining the correlation of a time series with its own values, shifted in time, backward and/or forward. The positive autocorrelation may be considered a special state of a persistent system in time, having the tendency to stay in the same state from one observation of the system to the next one. In practice, time series modeling geo-physics processes auto-correlate due to phenomena, such as inertia or carryover in the physical system [9].

This method may be used for identifying the signicative covariance or correlation between time-series. However, the most practical use of this analysis method is in forecasting, where it benefits from the properties of the auto-correlated time series. Since the future values of such a time series depend on the past ones, the series can be probabilistically predicted.

Having the measurements Y = (y_1, y_2, ... y_N) for the moments in time X = (x_1, x_2, ... x_N), where N is the total number of measurements on the time series, then the autocorrelation with the delay k (the correlation between observations separated by k years) is r_k, given by (1).



Search for a word: rocephin

Match count overview

Word usage chart

Year: 1986
Match pct: 1.725939191998274e-10
Norm pct: 0.036382880755664655

Figure 1. Example of the application output for the word rocephin.

$$r_k = \frac{\sum_{i=1}^{N-k}(y_i - \bar{y})(y_{i+k} - \bar{y})}{\sum_{i=1}^{N-k}(y_i - \bar{y})^2} \quad (1)$$

Although the sequence of moments in time when the measurements where made (X) is not explicitly used, it is assumed that the measurements where performed at equidistant moments in time.

Thus, the result of autocorrelation, the r_k coefficient, is obtained by correlating the same time series with the same values of y_i, but for different moments in time: x_i and x_{i+k}.

B. Dynamic Time Warping (DTW)

The DTW algorithm is well-known in many domains, being introduced in the 60s [10] and then intensely researched in the 70s in studies for voice recognition [11]. Nowadays, it is used in various domains, such as hand-writing recognition, gesture recognition, data mining and time-series clustering, computer vision, proteins alignment and chemical industry, music and signal processing, etc [12].

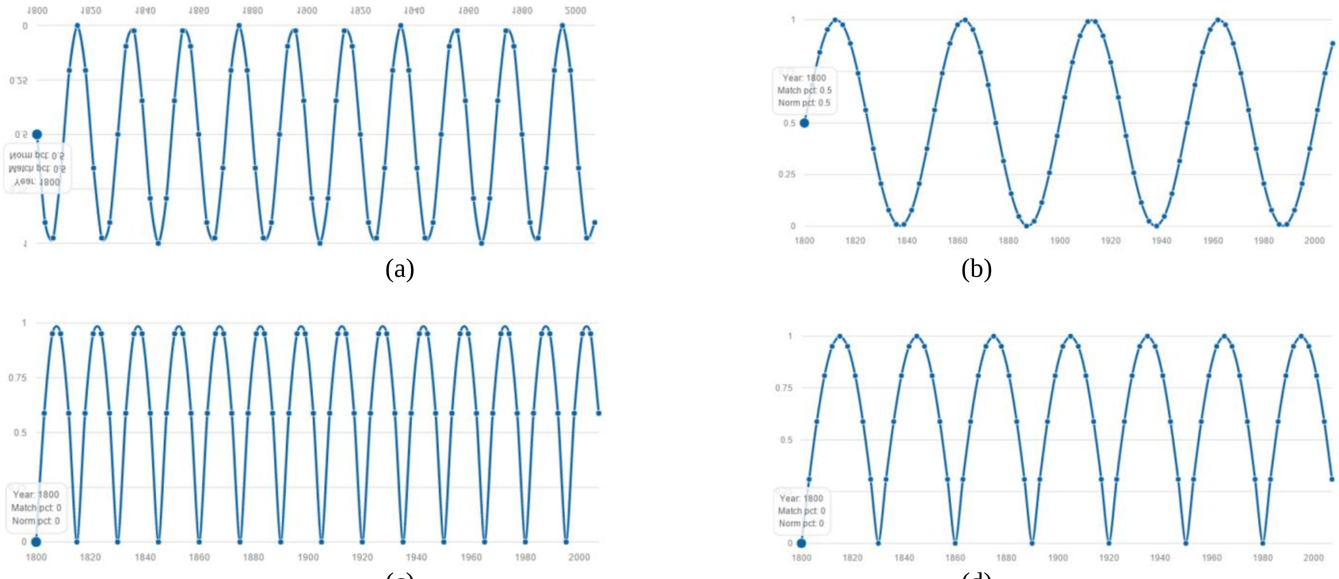
This algorithm earned its popularity due to its efficiency in recognizing the similarity between two time series, allowing an elastic transformation of the time series for detecting similar shapes.

Receiving at input two times series, X = (x_1, x_2, ... x_N), and Y = (y_1, y_2, ... y_M), with M, N ϵ $\mathbb{N}$, representing the values sequence of these time series, DTW computed the optimum solution with a complexity of O (M * N). The only restriction of this algorithm is that the series to be sampled at equidistant points in time.

In this research, we used the pseudo-code from [12] to implement the DWT algorithm for comparing some pre-defined time series with the ones obtained based on the words' usage over time. The pre-defined time series that we used had two different shapes: either sinusoidal or sinusoidal from which we maintained only the absolute values. We considered these shapes for the pre-defined time series, as these are cyclic (by definition) and thus, if a time series is similar to one of these, it means they are also cyclic. Besides varying the type of the pre-defined curves, we also varied their period, to allow the detection of cycles of various dimensions. Some examples of the used pre-defined curves can be seen in Figure 2.

V. Obtained Results and Discussion

Some of the obtained results can be visualized in Figure 3, while in Table 1 we present the words which yielded the best scores after running the algorithms for detecting cyclic words. As it can be seen, most of these words are part of the pharmaceutical domain.



(a)

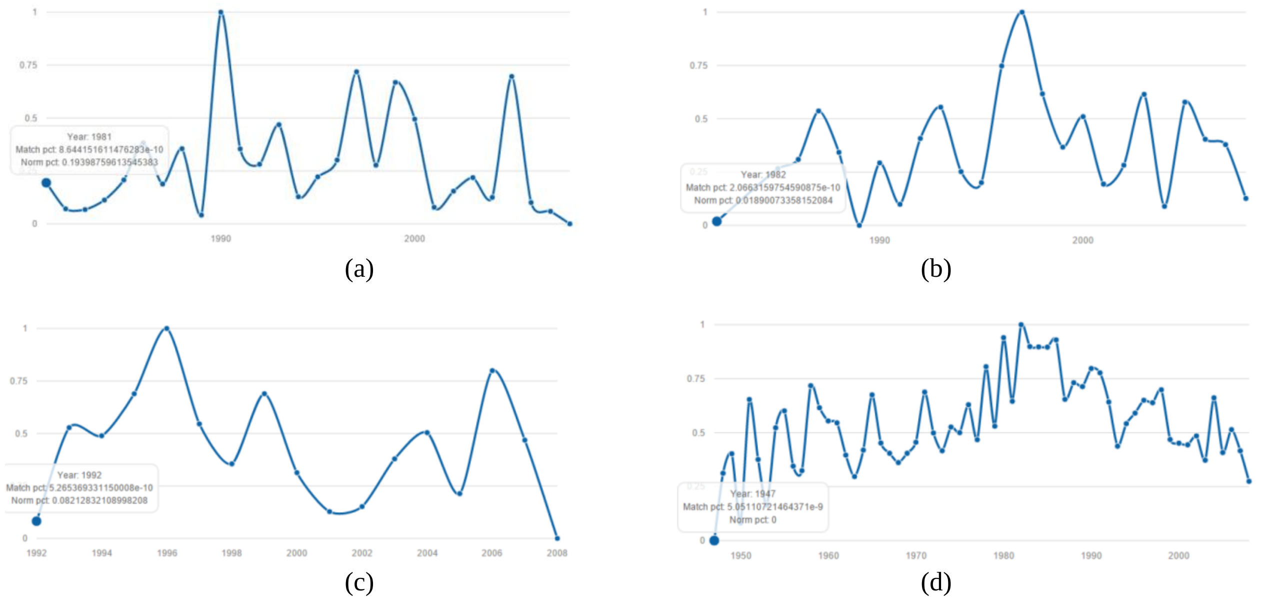
(b)

(c)

(d)

Figure 2. Pre-defined time series used in the DTW algorithm: (a) Sinusoid with a period of 20 years; (b) Sinusoid with a period of 50 years; (c) Sinusoid using only the absolute values having a period of 30 years; (d) Sinusoid using only the absolute values having a period of 60 years.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-563-0



(a)

(b)

(c)

(d)

Figure 3. The normalized graph for some of the identified cyclic words (a) anaprox; (b) augmentin; (c) didanosine; (d) propylthiouracil.

TABLE I. IDENTIFIED CYCLIC WORDS

Letter	Number of analyzed words	Detected cyclic words
A	2994	abacus, abdominoplasty, agave, aircrewman, allogeneic, alphanumerical, alphavirus, anaprox, anatomical, anticipation, ape
B	2241	basuco, beatrice, belief, bland, blarney, bobbysoxer, botch, brunt, brussels, buoyancy
C	4105	capacitive, catapres, clioquinol, codex, cognac, cognizant, collision, colonization, conceding, counterinsurgency, cowherd, cushion, cyberphobia
D	2446	dadaism, dbms, deathbed, decadron, decapitated, defunct, delavirdine, deoxythymidine, desertification, desyrel, didanosine, dislocate, dissect, domesticated, dronabinol
E	1808	egotrip, egyptologist, empennage, enalapril, enclosure, enthrall, eumycota, evergreen, excrement, extensively
F	1652	fainthearted, festering, fiddler, figment, fleshiness, frisian
G	1280	geological, gifted, glassy, gulf
H	1585	haldol, helmsman, herbaceous, hermes, hillbilly, history, honeycomb, horticultural, hydroxyzine, hyena, hypervolaemia
I	1875	illegible, immersion, inderal, induct, informercial, interlace, intralinguistic
J	371	joust
L	1506	lac, legitimately, leo, lifelessness, limnodromus, lindsay, linkup, llama, lopressor, lyophilise
M	2298	manifestation, marge, mentha, metricate, microelectronic, microphone, molehill, monosyllabic, montgomery, multiethnic, munro
N	876	nadolol, naltrexone, ncdc, nelson, neosporin, nonproliferation, nureyev, nydrazid
O	952	ominous, omnipresent, onerous, opponent, optative, oswald, outlandish, outpouring, overcome, overflight
P	3474	paedophile, paintbox, paramount, paternally, pectoralis, personify, pharmacogenetics, pimpled, plantago, plentitude, plop, polygonal, popular, postindustrial, privatize, propylthiouracil, psittacosaur, pyramid
R	1918	rarely, recoverable, reluctantly, remodel, renegade, resident, resoluteness, retrovirus, reverberating, ritalin, robertson, rocephin, roleplaying, root
S	4338	saquinavir, saturate, schtik, scott, scrutinise, seats, sectarianism, sedum, serratus, shoed, soliton, speaker, sporanox, sunchoke, supporter, swiss, switchblade
T	2127	teleconference, temp, theologian, tonocard, topicalization, toradol, tracing, transparence, tranylcypromine
U	1434	underboss, unfettered, unfinished, unimpeded
V	780	vacate, velban, videodisc
W	875	waking, willis, workings
Z	101	zinacef, zovirax

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-563-0

After applying the two algorithms, we compared the obtained results to highlight the differences between them, as well as each ones' advantages and disadvantages. Thus, both algorithms may be used for detecting if a graph varies regularly, but there are differences in what may be considered regular. Autocorrelation offers the best results when the graph has a shape that repeats at certain intervals, without imposing any restriction on the curve's shape. DTW algorithm compares the graph with a predefined shape. Thus, it detects that the time series varies regularly only if the two shapes are alike.

Thus, autocorrelation offers results that can be more generic, while DTW offers more specific ones, being required that the analyzed graph to be similar to the one imposed for comparison. From this observation comes the main advantage of autocorrelation, but also its main disadvantage. The advantage is given by the fact that the analyzed curves may have any shape, the only requirement being to vary regularly in order to autocorrelate. However, its disadvantage is generated by the fact that the graph may also autocorrelate when it is almost constant in time, with small variation that may be seen as noise. Therefore, the words whose use frequency stabilized in time will be identified by this algorithm as cyclic, generating inaccurate results.

VI. Conclusions

In this paper, we presented a system capable of indexing the unigram dataset provided by Google and of analyzing the graph of each indexed word. The analysis was done with the help of two different algorithms, autocorrelation and DTW, to establish if the graphic representation of the words' usage in time was cyclic or not.

As it can be seen in Figure 3 and in Table 1, most of the words whose use was cyclic in time are from the pharmaceutic domain. This may be attributed to the fact that the interest for pharmaceutic products (especially for the ones present in the results) tends to be sinusoidal, with ups and downs.

Finally, based on the obtained results, we cannot say which of the two algorithms is better, both having advantages as well as disadvantages. As already mentioned, autocorrelation has the advantage of identifying if a graph is cyclic, no matter what shape it has, but may end up with false autocorrelations in the case of constant use of a word. On the other hand, DTW has the advantage that to be cyclic means to have a sinusoidal shape, and thus it uses sinusoids to compare the words' usage graphs with. However, if the graph is cyclic but does not have a sinusoidal shape, then DTW will fail to identify it as cyclic.

In the future, we aim to continue the current research by grouping the cyclic words in clusters such as events, products, personalities, locations, sentiments, actions, etc. Thus, based on the word and its category, different conclusions may be drawn. For example, for cyclic words from the events cluster, one could predict when that type of event might happen or might become popular again, based on its cyclicity.

Acknowledgment

This work has been funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB – GEX. Identifier: UPB–EXCELENȚĂ–2016 Aplicarea metodelor de învățare automată în analiza seriilor de timp (Applying machine learning techniques in time series analysis), Contract number 09/26.09.2016.

References

[1] J.-B. Michel et al., Quantitative analysis of culture using millions of digitized books. Science, 331 (6014), pp. 176-182, 2011.

[2] A. M. Petersen, J. Tenenbaum, S. Havlin, and H. E. Stanley, "Statistical Laws Governing Fluctuations in Word Use from Word Birth to Word Death," Scientific reports, 2, Article number: 313, 2012, doi:10.1038/srep00313.

[3] S. Roth, "Fashionable Functions: A Google Ngram View of Trends in Functional Differentiation (1800-2000)," Int. J. of Technology and Human Interaction, 10(2), pp. 34-58, April-June 2014, doi: 10.4018/ijthi.2014040103.

[4] A. Acerbi, V. Lampos, P. Garnett, and R. A. Bentley, The Expression of Emotions in 20th Century Books, PloS one 8, no. 3, e59030, March 20, 2013

[5] A. Islam, E. Milios, and V. Kešelj, "Comparing Word Relatedness Measures Based on Google n-grams," Proceedings of COLING 2012: Posters, pp. 495–506, December 2012

[6] D. T. Wijaya and R. Yeniterzi, "Understanding Semantic Change of Words Over Centuries," Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web, pp. 35–40, ACM, 2011.

[7] G. A. Miller, WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, No. 11, pp. 39-41, 1995.

[8] G. E. P. Box and G. M. Jenkins, Time series analysis: forecasting and control, revised ed. Holden-Day, 1976.

[9] D. Meko. Autocorrelation. GEOS 585A: Applied Time Series Analysis, Course 3, The University of Arizona, 2005 [Online, accessed June, 2017]. Retrieved from: http://shadow.eas.gatech.edu/~jean/paleo/Meko_Autocorrelation.pdf

[10] R. Bellman and R. Kalaba, "On adaptive control processes," IRE Trans. on Automatic Control, vol. 4, no. 2, pp. 1–9, 1959.

[11] C. Myers, L. Rabiner, and A. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 6, pp. 623–635, 1980.

[12] P. Senin, "Dynamic time warping algorithm review," Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855, 1-23, 2008.