IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-4 E-ISSN: 2347-2693

Characteristic mining of Mathematical Formulas from Document - A Comparative Study on Sequence Matcher and Levenshtein Distance procedure

G. Appa Rao [1*], G. Srinivas [2], K.Venkata Rao [3], P.V.G.D. Prasad Reddy [4]

[1*] Department of CSE, GIT, GITAM,VISAKHAPATNAM,INDIA
[2] Department of IT, ANITS, VISAKHAPATNAM, INDIA
[3, 4] Department of CSSE, AndhraUniversity,VISAKHAPATNAM,INDIA

**Corresponding Author: gapparao@gmail.com*

Available online at: www.ijcseonline.org

Received: 20/Mar/2018, Revised: 28/Mar/2018, Accepted: 19/Apr/2018, Published: 30/Apr/2018

Abstract— The key predicament in the present circumstances is how to categorize the mathematically related keywords from a given text file and store them in one math text file. As the math text file contains only the keywords which are related to mathematics. The math dataset is a collection of huge amount of tested documents and stored in math text file. The dataset is trained with giant amount of text files and the size of dataset increases, training with various text samples. Finally the dataset contains only math-related keywords. The proposed approaches evaluated on the text containing individual formulas and repeated formulas. The two approaches proposed are one is Sequence matcher and another one is Levenshtein Distance, both are used for checking string similarity. The performance of the repossession is premeditated based on dataset of repetitive formulas and formulas appearing once and the time taken for reclamation is also measured.

***Keywords*—**Levenshtein distance,Sequence matcher

I. INTRODUCTION

The significance in mining of time series has been augmented in the recent years as it is very complex. Time series is very complex as there is high dimensionality, high affiliation between data. Majority of the technical papers are in print with mathematical formulas with time series. The search for related mathematical expression by the researchers is operational in technological discipline and cannot be done in effect with text based search engine except appropriate text keywords are known. The competence of reachable text search engine to search mathematical expression subjugated with a math-aware search engine [1, 2, 3].

The search for mathematical formulas is important and tricky as they restrain both constitutional and interpretation information. In general the theoretical basis of the knowledge in numerous technical documents is generally represented with mathematical formulas. The established search engines Google and Yahoo works competently for text based information. But they are besieged in searching data with mathematical formulas.
Let us consider an example Tanx+eSecx, this formula contains three symbols e, tan and sec signifies exponential and trigonometric functions. In this equation the tan and sec terms contains some semantic meaning and tan and sec terms are structurally associated to exponential.

As already quite a few methods were projected for the extraction of mathematical formulas, in this paper we planned two new methods for retrieving repetitive and non repetitive formulas materialized in the given text.

The later part of this paper is designed as follows. Section II reviews the related methods like Sequence Matcher and Levenshtein Distance used for the retrieval mathematical formulas. Section III deals with the function point. Section IV encompasses the comparative study of the proposed methods. As a final point, the conclusion is accessible in Section V.

II. RELATED WORK

A. Extracting Math formulas

Generally, Math formulae are written using MathML and Latex. The text file with Math Keywords is Loaded in our program in append mode. Then the math keywords from loaded text Document identified in Reading Mode. Pre-processing allows us to read the file and store every string in

the list and remove the stop words from the file. The next step is to load the math keyword document into the program and make a list which contains all the keywords related to math. Compare the math keyword list with the keywords in the text file to mine math keywords [4, 5, 6].

B. Sequence Matcher

Sequence Matcher is a flexible class for match up pairs of successions of any type, so long as the sequence elements are comparable. The Sequence matcher algorithm is little better than, an algorithm in print by Ratcliff and Obershelp under the name "gestalt pattern matching "in the late 1980's. The fundamental idea is to find the best ever contiguous matching subsequence that contains no useless elements. The same idea is then recursively applied to the portions of the sequences to the right and to the left of the matching subsequence. This does not capitulate insignificant edit sequences but does be inclined to capitulate matches that "look right" to people.

Sequence Matcher compares two strings and gives the ratio in between 0 to 1.When the comparison ratio between two strings is greater than 0.7 or 0.9 then it will be considered as keyword and keyword will be stored in the data set.

III. METHODOLOGY

A. Levenshtein Distance

The Levenshtein Distance is used to assess the resemblance between source string and objective string. The fundamental ideas of Levenshtein Distance are extensively used in areas like Computer Science, Computational Linguistics, Bioinformatics, Molecular Biology, DNA analysis. It can also be used in measuring the similarity of melodies or rhythms in music. The Levenshtein distance has widely pervaded in our day to day life. Levenshtein distance or edit distance is used for spell checking and error correction in a program or an application. Another possible use of the Levenshtein distance is in speech recognition and Plagiarism detection.

The amalgamation of Levenshtein distance or edit distance with Trie index locates related words faster. In order to transform one word into another word Levenshtein distance refers to the number of single character operations such as insertion, replacement or deletion. The edit distance between “pen” and “hen” is one, since substituting the character ‘p’ by ‘h’ the word “pen” can be converted to “hen”. The threshold for edit distance is determined based on the length of the all similar words within the threshold distance.

B. Function Point

The math keyword identified from the file will be stored in keywords dataset, if the identified key word is not available in the data set. The latest dataset is formed with the newly appended keywords identified. The new data set is used to test any other file. We trained around more than 20 documents to our program and different keywords identified from it are stored in the dataset. For every document, the newly formed keywords are appended to the dataset. At last, we have huge dataset contain all math related keyword from the testing of various samples. In proposed approaches the time between the document loading and attaining appropriate results is calculated.

IV. RESULTS AND DISCUSSION

A. Comparative study

The performance of Sequence Matcher and Levenshtein Distance are measured in this section. Initially there is only one dataset file that is used for string comparison, after preprocessing and executing with sequence matcher a new dataset will be obtained. The size of the data set will increase after successive string comparisons. The second method called Levenshtein Distance for string comparison is base on single character operations. The better the Levenshtein distance, the more dissimilar the strings are. In order to find out the efficiency of proposed models around 20 different samples of documents were used. The proposed approaches identify math related keywords along with some unwanted words. If any math related keyword is present in the tested document it is dynamically inserted into our math dataset if it does not exist before, next time while loading file the newly updated dataset is used for string matching. The efficiency of proposed approaches is measured in terms of time analysis. String comparison with sequence matcher is completed with less time than Levenshtein distance. But rarely in some cases, does Levenshtein distance gives accurate results when compared to sequence matcher.

B. Time Analysis of two proposed approaches

Table 1: Time for retrieving matched formulas from the document with more number plus symbol with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
40	3.042	3.28

Table 2: Time for retrieving matched formulas from the document with more number of Algebraic symbols with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	3.046	2.91

Table 3: Time for retrieving matched formulas from the document with more number of Arithmetic symbols with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	3.23	3.34

Table 4: Time for retrieving matched formulas from the document with more number of Cube symbols with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	7.23	3.35

Table 5: Time for retrieving matched formulas from the document with more number of Differentiation formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	**2.70**	**6.76**

Table 6: Time for retrieving matched formulas from the document with more number of Exponential formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.61	2.84

Table 7: Time for retrieving matched formulas from the document with more number of Integral formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.74	17.32

Table 8: Time for retrieving matched formulas from the document with more number of Limit formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.75	15.42

Table 9: Time for retrieving matched formulas from the document with more number of Logarithmic formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
40	5.59	2.62

Table 10: Time for retrieving matched formulas from the document with more number of Pi formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	3.09	3.22

Table 11: Time for retrieving matched formulas from the document with more number of Square Root formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	3.03	2.97

Table 12: Time for retrieving matched formulas from the document with more number of Sigma formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.74	3.84

Table 13: Time for retrieving matched formulas from the document with more number of Square formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	15.2	2.80

Table 14: Time for retrieving matched formulas from the document with more number of Subtraction formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.82	2.93

Table 15: Time for retrieving matched formulas from the document with more number of Trigonometric formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
20	2.80	17.3

Table 16: Time for retrieving matched formulas from the document with more number of Combination formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	3.21	3.24

Table 17: Time for retrieving matched formulas from the document with more number of Factorial formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	4.23	3.39

Table 18: Time for retrieving matched formulas from the document with more number of Permutation formulae with sequence matcher and Levenshtein Distance

Number of formulae	Sequence Matcher	Levenshtein Distance
30	4.62	2.95

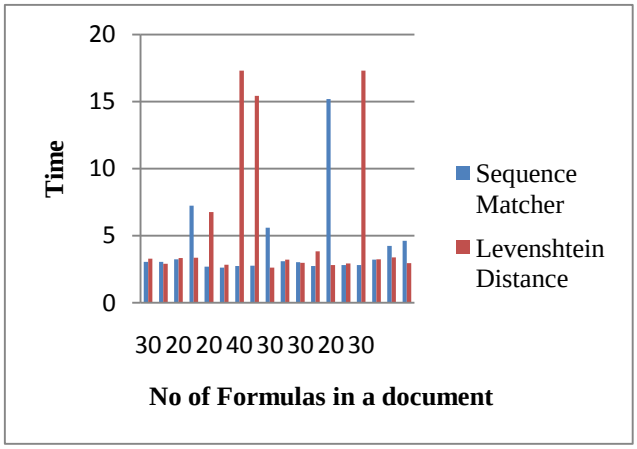

Fig 1: Comparison between Sequence Matcher and Levenshtein Distance in terms of Time analysis.

V. CONCLUSION and Future Scope

In this paper we projected two approaches which retrieve repeated and non repeated mathematical formulas. The recitation of the planned methods measured in terms of time analysis. The sequence matcher technique does not retrieve all the strings. But Levenshtein Distance retrieves some unwanted strings along with the matched strings. The Sequence matcher matches the string in less time than Levenshtein Distance. Presently experiments are being performed on much enormous set of training documents. Currently we are working with a method called fuzzy-wuzzy by using this we can get accurate results when compared to Sequence matcher and Levenshtein Distance.

.

REFERENCES

[1] Kai Ma, Siu Cheung Hui and Kuiyu Chang "*Feature Extraction and Clustering-based Retrieval for Mathematical Formulas",* pp. 372-377.

[2] Sidath Harshanath Samarasinghe and Siu Cheung Hui *"Mathematical Document Retrieval for Problem Solving",* International Conference on Computer Engineering and Technology, pp.583-587,2009.

[3] J. Misutka and L. Galambos, *"Mathematical Extension of Full Text Search Engine Indexer",* Proc. 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 08), , pp. 1-6,April 2008.

[4] B.R. Miller and A. Youssef, *"Technical Aspects of the Digital Library of Mathematical Functions",* in Annals of Mathematics and Artificial Intelligence, Springer Netherlands, pp. 121-136, 2003.

[5] H. Zhang, T.B. and M.S. Lin, *"An Evolutionary Kmeans Algorithm for Clustering Time Series Data"* ,Proc. International Conference on Machine Learning and Cybernetics, pp. 1282-1287, 2004.

[6] R. Munavalli and M.R. MathFind, *"A Math-aware Search Engine",* Proc. Annual International ACM SIGIR Conference on Research and development in information retrieval, pp.735-735, 2006.

[7] M. Kohlhase. *"Markup for Mathematical Knowledge," An Open Markup format for Mathematical Documents",* Ver. 1.2, Lecture Notes in Computer Science, , Springer Berlin, pp. 13-23.

[8] G.AppaRao,K.Venkata Rao,PVGD Prasad Reddy and T.Lava Kumar,*"An Efficient Procedure for Characteristic mining of Mathematical Formulas from Document",* International Journal of Engineering Science and Technology (IJEST), Vol. 10 No.03,pp152-157, Mar 2018

Authors Profile

Mr G.Appa Rao obtained his B.Tech, Computer Science & Engineering,and M.Tech, in Computer Science & Technology.He is pursuing Ph.D. and currently working as Assistant Professor in the Department of Computer Science & Engineering,Gitam Institute of Technology,GITAM,since 2007. His main research work focuses on Text Based Mining,and Soft Computing. He has 14 years of teaching experience.

Dr G.Srinivas, obtained his M.Tech, in Computer Science & Technology and Ph.D in Computer Science & Engineering.He is currently working as Associate Professor in the Department of Information Technology in ANITS.His area of research is Image Processing. He has 14 years of teaching and industry experience.

Prof.K.Venkata Rao,obtained his B.Tech,in Computer Science and Engineering and M.Tech in Science & Technology and Ph.D in Computer Science and Engineering. He is presently Professor in Computer Science & Systems Engineering department and Honorary Director for the Computer Centre. He has 21 years of teaching & research experience His Research areas include Image Processing,Internet Technologies and Web & Cyber Security.

Prof. Prasad Reddy, P.V.G.D, obtained his B.Tech, in Mechanical Engineering, from Andhra University and M.Tech, in Computer Science & Technology, and Ph.D, in Computer Engineering. He is presently the Head of Computer Science & Systems Engineering department.He has to his credit, more than 120 Research papers which includes around 95 publications in the referred indexed Journals.His Research areas include Soft Computing, Software Architectures, knowledge Discovery from Databases, Image Processing, Number theory & Cryptosystems.He has 24 years of teaching & research experience which includes more than 10 years of administrative experience with Andhra University at various senior capacities.

