Information Retrieval from Web Documents using Pattern Matching Algorithms

[*]Mrs. R. Janani, [#]Dr. S. Vijayarani

[*]Ph.D. Research Scholar, [#]Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, India. [*]janani.sengodi@gmail.com, [#]vijimohan_2000@yahoo.com

Abstract - Web mining is the key area of data mining, it is used to detect patterns and extract useful information from web documents and web services. Web mining is classified into web usage mining, web content mining and web structure mining. This research work mainly focused on web content mining. It is used to extract the important and useful data, information and knowledge from the contents of the web pages. It explains the discovery of useful information from the collection of web pages. In web content mining the contents may be a text, image, audio, video, metadata and hyperlinks. The main application area of web content mining is document clustering, document classification and information extraction from the web pages. In this research work, pattern matching algorithms are used for web page content analysis and these algorithms are used to match the pattern exactly. The main objective of this research work is to retrieve the relevant information from a collection of web documents. For this analysis, two algorithms are used; they are Turbo BM algorithm and Backward Nondeterministic Dawg Matching (BNDM) algorithm. From this analysis, based on the performance measures it is determined that the Backward Nondeterministic Dawg Matching (BNDM) algorithm gives the better result.

Keywords: Web mining, pattern matching, content mining, Turbo BM algorithm, Backward Nondeterministic Dawg Matching (BNDM) algorithm

I. INTRODUCTION

Web mining is the application of data mining techniques to extract knowledge from web data, which comprises web documents, hyperlinks between documents, and usage logs of web sites [1]. This research work mainly focused on web content mining. For analyzing the web page content pattern matching algorithms are used. Pattern matching algorithms are necessary for many problems and it is used in various applications which includes text mining, data retrieval, DNA pattern matching and finding certain important keywords in the security applications [2]. This algorithm has two strategies such as, exact matching and approximates matching. In exact matching, the pattern is completely matched with the specific text window of input text and it displays the initial index position [3]. In approximate matching, if precise portion of the pattern is matched with the selected text window straight away it displays the output.

There are different kinds of string or pattern matching tools accessible and those tools will provide partial or approximate of string/pattern matching results. Though some of the tools are based on exact matching of the given strings/patterns. But, the total number of sequences is increasing rapidly, to handle this situation the efficient methods are to be proposed.

This paper is organized as follows. Section II describes the related works; Section III illustrates the methodology of the research work. Result and discussions are given in Section IV and section V gives the conclusion of this research work.

II. RELATED WORKS

Sathish Kumar et al. [1] presented the research in the area of applications of neural networks and pattern matching algorithms in classification. Artificial neural networks for classification and different pattern matching algorithms are used for matching the given DNA patterns or strings with the existing DNA sequences available in the databases are specially studied. A number of indigenous pattern matching algorithms were investigated for different test string lengths and their time difficulty is tabulated.

Pandiselvam.P et al. [4] discussed the functional and structural relationship of the biological sequence which is determined by similarities on that sequence. So that, the researcher is supposed to aware of similarities on the biological sequences. In this research work authors have studied different kinds of string matching algorithms and observed their time and space complexities. For this study, they have assessed the performance of algorithms by testing with some of the biological sequences.

Abdulwahab Ali Al-mazroi et al. [5] proposed a new hybrid algorithm called BRSS by combining two algorithms, Berry-Ravindran and Skip Search. The hybrid algorithm demonstrates enhanced character comparisons, number of attempts and searching time performances in all the

 DOI : 10.18231/2454-9150.2019.0228

different data size and pattern lengths, therefore the proposed algorithm is useful for searching DNA, Protein and English text. This also proved that the application of the hybrid algorithm will lead to better searching and matching of the patterns than the existing pattern matching algorithm.

Saima Hasib et al. [6] discussed the Aho-Corasick algorithm is best suited for multiple pattern matching and it can be used in many application areas. The complexity of the algorithm is linear in the length of the patterns plus the time taken for the searched text plus the amount of output matches. It is establishing to be attractive in huge numbers of keywords, since all keywords can be concurrently matched in one pass. Aho-Corasick affords solution to a number of real world problems like Intrusion detection, Plagiarism detection, bioinformatics, digital forensic, text mining and many more. Aho-Corasick is one of the most inventive algorithms in text mining.

Jorma Tarhio et al. [7] proposed an efficient string matching algorithm (named ACM) with solid memory as well as high worst-case performance. By means of a magic number experimental based on the Chinese Remainder Theorem, the proposed ACM suggestively diminishes the memory prerequisite without bringing the complex processes. Moreover, the latency of off-chip memory situations is extremely reduced. The proposed ACM can be easily instigated in hardware and software. As a result, ACM enables cost-effective and efficient IDSs.

Chinta Someswara Rao et al [8] implemented parallel string matching with JAVA Multithreading with multi core processing, and performed a comparative study on Knuth Morris Pratt, Boyer Moore and Brute force string matching algorithms. For testing, gene sequence database is used which consists of lacks of records. From the test results it is shown that the multi core processing is better compared to lower versions. Finally, this proposed parallel string matching with multi core processing is better compared to other sequential approaches.

III. METHODS

The main objective of this research work is to retrieve the relevant information from a collection of web documents. In order to perform this task, this research work uses two phases; Pre-processing phase and Searching phase. In the preprocessing phase the converter has to be used to convert the web documents into pdf file format. In the searching phase, two pattern matching algorithms are used namely Turbo BM algorithm and Backward Nondeterministic Dawg Matching (BNDM) algorithm. The performance factors are time taken for searching the pattern, number of iterations and the relevancy.

A. Web Documents

In order to perform this task, the input documents are collected from the web. For this analysis fifteen web document links are used as input.

B. Preprocessing Phase

In this phase the converter is used for converting web documents into pdf file format. The content analysis has to done in the pdf file format [9]. The name of the converter is WebpagetoPDF. It is the open source online tool, and it helps the web users, website publishers and bloggers to save their web content to pdf files, so that they can print, share and archive the web documents and manipulating the important web contents. It is easy to use, users can just copy and paste the URL of the page they want to save and click the convert button. This tool also gives the options for users like gray scale, landscape, low quality, no background and no java script. Table 1 describes the original web pages given into the Webpage to PDF and it converts the web pages into pdf files. The converted pdf with their names is given in Table 2.

TABLE 1: SAMPLE INPUT

S. No	Web Document Links
1	http://www.merriam-webster.com/dictionary/multimedia
2	https://www.cics.umass.edu/admissions/healthcare
3	https://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.htm
4	http://hortonworks.com/apache/hdfs/
5	https://en.wikipedia.org/wiki/Parallel_algorithm
6	http://wikid.eu/index.php/Image_Mining
7	http://www.tcs.com/SiteCollectionDocuments/White%20Papers/Insurance_Whitepaper_Mining_Unstructured_Text_Data_for_Insurance_Analytics_08_2010
8	https://www.linguamatics.com/blog/text-mining-full-text-scientific-articles-more-facts-more-types-facts-faster
9	http://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining
10	http://www.sciencedirect.com/science/article/pii
11	https://en.wikipedia.org/wiki/Document_clustering
12	https://en.wikipedia.org/wiki/Web_mining
13	https://en.wikipedia.org/wiki/Data_mining
14	https://en.wikipedia.org/wiki/Social_media_mining
15	http://scikit-learn.org/stable/auto_examples/text/document_clustering.html

Figure 1: Home Page of Converter

 DOI : 10.18231/2454-9150.2019.0228

TABLE 2: PDF CONVERSION OF WEB PAGES – PDF NAMES

S. No	Web Document Links	Converting the Link to Pdf using Web2PDF
1	http://www.merriam-webster.com/dictionary/multimedia	merriam-webster_com.pdf
2	https://www.cics.umass.edu/admissions/healthcare	umass_edu.pdf
3	https://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.htm	tutorialspoint_com.pdf
4	http://hortonworks.com/apache/hdfs/	Hortonworks_com.pdf
5	https://en.wikipedia.org/wiki/Parallel_algorithm	wikipedia_org%20Par.pdf
6	http://wikid.eu/index.php/Image_Mining	wikid-eu.pdf
7	http://www.tcs.com/SiteCollectionDocuments/White%20Papers/Insurance_Whitepaper_Mining_Unstructured_Text_Data_for_Insurance_Analytics_08_2010	Tcs_com.pdf
8	https://www.linguamatics.com/blog/text-mining-full-text-scientific-articles-more-facts-more-types-facts-faster	linguamatics_com.pdf
9	http://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining	techtarget_com.pdf
10	http://www.sciencedirect.com/science/article/pii	sciencedirect_com.pdf
11	https://en.wikipedia.org/wiki/Document_clustering	wikipedia_org%20Doc.pdf
12	https://en.wikipedia.org/wiki/Web_mining	wikipedia_org%20Web.pdf
13	https://en.wikipedia.org/wiki/Data_mining	wikipedia_org%20Dat.pdf
14	https://en.wikipedia.org/wiki/Social_media_mining	wikipedia_org%20Soc.pdf
15	http://scikit-learn.org/stable/auto_examples/text/document_clustering.html	scikit-learn_org.pdf

C. *Searching Phase*

In this phase the pattern matching algorithms are used to search the particular pattern in the text [11]. For this analysis there are two algorithms are used namely Turbo BM algorithm and Backward Nondeterministic Dawg Matching (BNDM) algorithm.

D. *Turbo-BM Algorithm*

The Turbo BM algorithm is developed from the Boyer-Moore algorithm. This algorithm needs no extra preprocessing but there is a need for extra space which is added with respect to the original Boyer-Moore algorithm [12]. In this algorithm, we remember the last time suffix pattern which was matching. It has two main advantages, they are, it is potential to jump over factor and it may enable to perform a turbo shift operation. In this algorithm, the preprocessing phase can be done in O (m+σ) and the searching phase can be done in O (n) time complexity. The number of text character comparisons performed by the Turbo-BM algorithm is limited by 2n [12].

Algorithm 1: Turbo BM Algorithm

Step 1: Let i is equal to 0 and memory allocated is nil, p is the position;
Step 2: while i ≤ n-m do
{
begin
align pattern with positions t[i + 1 . . i + m] of the text;
scan the text right-to-left from the position i + m, using memory to reduce number of inspections;
}
Step 3: Let x be the part of the text scanned;
Step 4: if (x = p)
{
report a match at position i;
compute the shift shift i according to x and memory;
take i as i + shift i;
update memory using x;
}

E. *Backward Nondeterministic Dawg Matching algorithm*

It is the variant of the Reverse Factor algorithm. It uses bit parallelism simulation of the suffix automaton of x^R. This algorithm is efficient if the pattern length is no longer than the memory-word size of the machine. The automaton is simulated with bit parallelism even without constructing it. In this algorithm, for each character, associated tables B will be precompiled with a bit mask expressing its occurrences [13].

Algorithm 2: Backward Nondeterministic Dawg Matching Algorithm

Step 1: Let m is the memory size, and initialize i, j, d, last.
Step 2: If (m > word size)
{
Error
}
Step 3: Preprocessing Phase
Step 3.1: begin
{
m (B,0,arraysize *sizeof(integer))
initialize s to 1
for i = m-1 to i ≥ 0 do
{
B(x(i)) = B(x(i)) or S
left shift s by one bit
}
Step 4: Searching Phase
Step 4.1: Initialize j to 0
While j<=n-m
{
begin i=m-1;
last=m;
d is not equal to 0
while(i>=0 and d !=0)
{
begin
d EQUALTO d and s(y(j+1));
i=i-1;
if (d != 0)
{
begin
if i >= 0 last=i+1;
else output(j);
}
left shift d by one bit if
}
j=j+last
}

IV. RESULT AND DISCUSSION

In order to perform this analysis, the performance factors are search time, number of iterations and relevancy for various types of inputs. The inputs are single word, multiple

 DOI : 10.18231/2454-9150.2019.0228

words and a file for pdf file formats. For this analysis, the pattern matching algorithms were implemented by using Java. Here the input query is "Mining", "Text Mining", "Text Mining is also text data mining" for single word, multiple words and file respectively.

- Search Time: It refers to the time taken for searching the pattern within the input text. It can be estimated by comparison of each character in pattern with the input text.
- Iterations: It refers to the total number of iterations for matching the pattern with the input text. It is based on the given input document and various algorithms.
- Relevancy: It refers to the accuracy of the algorithm; the accuracy is calculated by using the formula as follows,

$$\text{Accuracy} = \frac{\textit{Total number of patterns retrieved}}{\textit{Total number of patterns in text}} \text{ X } 100$$

Table 3 shows the sample input used for this experimentation and the size of each files.

TABLE 3: SAMPLE INPUT WITH SIZE

File Name	Size (Kb)
merriam-webster_com.pdf	656
umass_edu.pdf	338
tutorialspoint_com.pdf	179
Hortonworks_com.pdf	982
wikipedia_org%20Par.pdf	496
wikid-eu.pdf	91
Tcs_com.pdf	736
linguamatics_com.pdf	81
techtarget_com.pdf	416
sciencedirect_com.pdf	48
wikipedia_org%20Doc.pdf	151
wikipedia_org%20Web.pdf	198
wikipedia_org%20Dat.pdf	481
wikipedia_org%20Soc.pdf	271
scikit-learn_org.pdf	377

Table 4 describes the performance analysis of Turbo BM algorithm and Backward Nondeterministic Dawg Matching (BNDM) algorithm.

TABLE 4: PERFORMANCE ANALYSIS OF TURBO BM AND BACKWARD NONDETERMINISTIC DAWG MATCHING (BNDM) ALGORITHM.

Input Pattern	Turbo BM Algorithm			Backward Nondeterministic Dawg Matching (BNDM) Algorithm		
	Time (ms)	Number of Iterations	Relevancy (%)	Time (ms)	Number of Iterations	Relevancy (%)
Single Word	04	11	100	06	10	100
Multiple Words	25	29	100	21	27	100
File	61	31	95	49	18	97

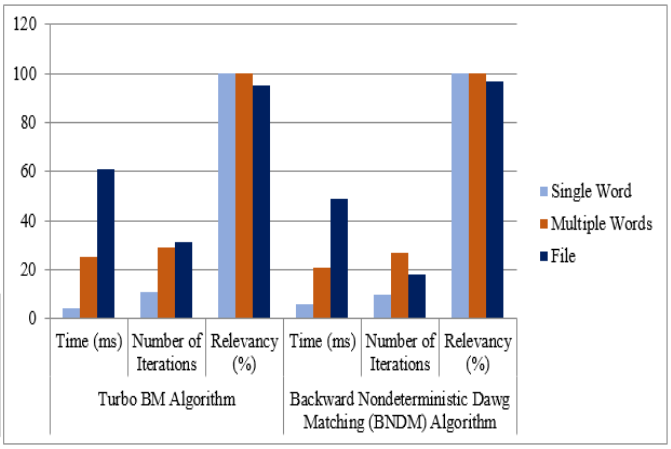

Figure 2: Performance analysis of Turbo BM and Backward Nondeterministic Dawg Matching (BNDM) Algorithm

Figure 2 illustrates the performance analysis of Turbo BM and Backward Nondeterministic Dawg Matching (BNDM) Algorithm. From this figure it is observed that Smith algorithm gives better result. Table 5 illustrates the ranking the documents based on pattern occurred in the particular document.

In Figure 3, the ranking of documents was given. It is based on how frequently the input patterns such as, Mining, Text Mining and a file occurred in the experimental documents.

TABLE 5: RANKING THE DOCUMENTS BASED ON PATTERN OCCURRED IN THE PARTICULAR DOCUMENT

File Name	Total Number of words	Total number of times Pattern occurred	Rank
merriam-webster_com.pdf	968	1	13
umass_edu.pdf	343	2	12
tutorialspoint_com.pdf	930	10	7
Hortonworks_com.pdf	1369	3	11
wikipedia_org%20Par.pdf	2603	48	4
wikid-eu.pdf	956	10	7
Tcs_com.pdf	5680	66	3

 DOI : 10.18231/2454-9150.2019.0228

linguamatics_com.pdf	867	6	10
techtarget_com.pdf	978	12	6
sciencedirect_com.pdf	594	8	9
wikipedia_org%20Doc.pdf	1194	0	15
wikipedia_org%20Web.pdf	2642	71	2
wikipedia_org%20Dat.pdf	6118	109	1
wikipedia_org%20Soc.pdf	2931	28	5
scikit-learn_org.pdf	1041	1	13

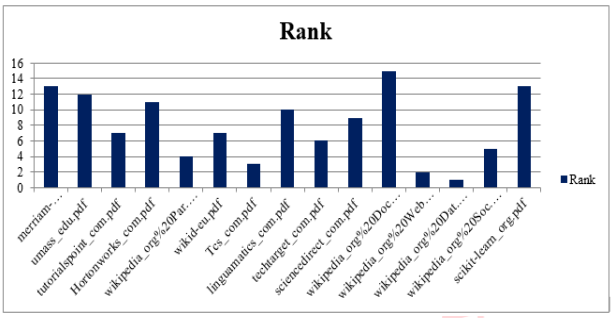

Figure 3: Ranking of Documents

V. CONCLUSION

Web mining is used to collect, organize and provide the precise information on the web based on the user's query. It is used to decide the relevance of the content to the search query. This research work mainly focused on web content mining, it uses the ideas of data mining. For mining or analyzing the web content, the pattern matching algorithms are used in this research work. The main objective of this work is to retrieve the relevant information from the web pages. In order to perform this task, there are two pattern matching algorithms are used. From the experimental results it is clearly observed that the Backward Nondeterministic Dawg Matching (BNDM) algorithm performs well in terms of accuracy. Also it extracts the relevant patterns from the documents based on user query.

REFERENCES

[1]. Sathish Kumar S and N. Duraipandian , Artificial Neural Network based String Matching Algorithms for Species Classification – A Preliminary Study and Experimental Results, International Journal of Computer Applications (0975 – 8887) Volume 52– No.14, August 2012

[2]. Mahmoud Moh'dMhashi , Mohammed Alwakeel, New Enhanced Exact String, Searching Algorithm, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010

[3]. Christian Charras, Thierry Lecroq and Joseph Daniel, A Very fast string searching algorithm for small alphabets and long patterns, Combinational Pattern Matching, 9th Annual Symposium, CPM 98 Piscataway, New Jersey, USA, 2005

[4]. Pandiselvam.P, Marimuthu.T, Lawrance. R, A Comparative Study On String Matching Algorithms Of Biological Sequences

[5]. R.S. Boyer, J.S. Moore, "A fast string searching algorithm," Communication of the ACM, Vol. 20, No. 10, 1977, pp.762– 772.

[6]. Abdulwahab Ali Al-Mazroi and Nur'aini Abdul Rashid, A Fast Hybrid Algorithm for the Exact String Matching Problem, American Journal of Engineering and Applied Sciences 4 (1): 102-107, 2011.

[7]. Ababneh Mohammad, OqeiliSaleh and Rawan A Abdeen, Occurrences Algorithm for String Searching Based on Brute-Force Algorithm, Journal of Computer Science, 2(1): 82-85, 2006.

[8]. Raju, S. V., Reddy, K. K. V. V. S., & Rao, C. S. (2018). Parallel String Matching with Linear Array, Butterfly and Divide and Conquer Models. Annals of Data Science, 1-27.

[9]. Bin Wang, Zhiwei Li, Mingjing Li and Wei-Ying Ma, Large-Scale Duplicate Detection for Web Image Search, Multimedia and Expo, IEEE International Conference, 353-356, 2006

[10]. JormaTarhio and EskoUkkonen, Approximate Boyer-Moore String Matching, SIAM Journal on Computing, Volume 22 Issue 2, 243 – 260, 1993.

[11]. Ashish Prosad Gope , Rabi Narayan Behera, A Novel Pattern Matching Algorithm in Genome Sequence Analysis, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5450-5457

[12]. Michailidis, P. D., & Margaritis, K. G. (2001). On-line string matching algorithms: Survey and experimental results. International journal of computer mathematics, 76(4), 411-434.

[13]. Lecroq, T. (2007). Fast exact string matching algorithms. Information Processing Letters, 102(6), 229-235.

 DOI : 10.18231/2454-9150.2019.0228