International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 6 1440 – 1442

Study on Distance Measures for Clustering of Web Documents based on DOM-Tree based Representation of Web Document Structure

Manoj Kumar Sarma
Department of Computer Science
Gauhati University
Guwhaati-781014, Assam, India
e-mail: manojksarma@yahoo.com

Anjana Kakoti Mahanta
Department of Computer Science
Gauhati University
Guwhaati-781014, Assam, India
e-mail:anjanagu@yahoo.co.in

Abstract—Among the three broad areas of Web mining, Web Structure Mining is the method of discovering structure information from either the web hyperlink structure or the web page structure. In order to apply data mining techniques on web pages, a good and efficient representation of web pages is required that could depict the actual hierarchical structure of web pages. The work presented here aims to find out an appropriate distance measure (also called as similarity measure) for strings that can be used for clustering of web documents and also for other data mining applications.

Keywords-*WebMining; String Encoding, Similarity measure, Distance measure.*

*****

I. INTRODUCTION

Web mining [1] is a special domain of data mining that is defined as the application of data mining techniques in order to mine knowledge from web data. The attention received by web mining from research, IT industry, and several web-based organizations has acquired significant experience.

Web mining [6] is a special domain of data mining that is defined as the application of data mining techniques in order to mine knowledge from web data. The attention received by web mining from research, IT industry, and several web-based organizations has acquired significant experience.

Web mining has three broad sub-areas- web content mining, web structure mining, and web usage mining.

Web content mining is the method of extracting useful information from the contents of web documents, that may contain text, images, audio, video, lists, tables. Application of text mining to web content is one of the emerging research areas. Web structure mining is the method of discovering structure information from either the web hyperlink structure or the web page structure. Web usage mining is the use of data mining techniques so as to discover web usage patterns from web usage data, generally stored in web server logs.

Applications of Web mining [9,10] include developing intelligent web search engines, developing e-com (business/auction) sites based on customer preference, navigation patterns, and behavior (especially B2C commerce), understanding Web communities, personalized Portal for the Web, load balancing on web servers, Digital Library and Autonomous Citation Indexing, etc.

Clustering algorithms [5,7]aim at partitioning data into a number of clusters (groups, subsets, or categories). A cluster is generally described by considering the internal homogeneity and the external separation. It is expected that patterns in a particular cluster should be similar to each other, whereas the patterns in two different clusters should not.

The four basic steps in Clustering [5] are- Selection or extraction of features, design or selection of a clustering algorithm, validation of clusters, & interpretation of results.

Different types of Clustering Algorithms include [7]- Partitioning Methods (e.g. K-means, PAM, CLARA, CLARANS), Hierarchical Methods (e.g. ROCK, QROCK, BIRCH, CURE), Density-Based Methods, (e.g. DBSCAN, OPTICS, DENCLUE), Grid-Based Methods (e.g. STING, CLIQUE) and Model-Based Methods (e.g. COBWEB, SOM).

There are many available representations and clustering algorithms for text documents. Although widely used, the Vector-Space Model does not preserve the order of the words. Hence it is not suitable for web documents.

All clustering algorithms are based on the calculation of similarity measures or distance measures between objects. However, distance measures that are used for numeric data are not applicable for string data.

II. LITERATURE REVIEW AND MOTIVATION

M. J. Zaki [2] in July 2002 proposed a string representation Tree data structures. The author pointed out that in contrast to the standard ways of tree representation, viz. an adjacency matrix or adjacencylist. the adopted string representation of a tree is more versatile and that could be applicable for efficient subtree counting and manipulation. He also pointed out that the representation could be extended to tree-structured documents like HTML or XML documents with proper customization for various web mining applications.

KabitaThaoroijam [7] in 2011 in her thesis has presented Text Document Clustering Using a Fuzzy Representation of Clusters. She adopted a two-phase agglomerative approach with the QROCK (Quick ROCK) algorithm to efficiently cluster text documents.The model used for representation of text documents was the Vector Space Model [11]. In Chapter 2 of her thesis, she has presented a comparative study of several similarity measures based on Vector Space model-based representation of text documents. She has also mentioned that the choice of similarity or distance measure is an important step in any clustering algorithm, because the choice drastically affects the clustering quality.

Since the present works aims at applying a string representation of web documents or web document clustering,

there become a need to find out an appropriate similarity or distance measure for strings [4,8].

The Stringdist [13] package in R includes Implementation of an approximate string matching version of native 'match' function available in R. The available functions can be used to calculate various string distances based on edits (Damerau-Levenshtein, Hamming, Levenshtein, optimal sting alignment), qgrams (qgram, cosine, jaccard distance) or heuristic metrics (Jaro, Jaro-Winkler).

The Stringdist package also includes an implementation of soundex. It is significant here that the distances can be computed between character vectors while taking proper care of encoding or between integer vectors representing generic sequences.

The stringdist package aims to offer fast and platform-independent string metrics. The main purpose of thhe package is to compute various string distances and to do approximate text matching between character vectors.

III. Background of The Proposed Work

A. *Available Methods:*

Available Methods:

1. The **Optimal String Alignment** distance (method='osa'): Allows transposition of adjacent characters. Each substring may be edited only once.
2. The **Levenshtein**distance (method='lv'): Counts the number of deletions, insertions and substitutions necessary to turn string b into string a.
3. The full **DamerauLevensthein** distance (method='dl'): like the optimal string alignment distance except that it allows for multiple edits on substrings.
4. The **Longest Common Substring** (method='lcs'): Based on the longest string that can be obtained by pairing characters from string a and string b while keeping the order of characters intact. The lcs distance is defined as the number of unpaired characters.
5. The **qgram** (method='qgram'): Based on a subsequence of q consecutive characters of a string.
6. The **Cosine** distance (method='cosine'): computed as COSINE of vectors representing strings a and b, as x.y/|x||y|
7. The **Jaccard**distance (method='jaccard'): Given by S_J = a/(a + b + c), where S_J = Jaccard similarity coefficient, a = number of species common to (shared by) quadrats, b = number of species unique to the first quadrat, and c = number of species unique to the second quadrat
8. The **Jaro** distance (method='jw', p=0): The Jaro distance of two given strings s_1 and s_2 is

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

Where ***m*** is the number of matching characters, ***t*** is half the number of transpositions.

IV. Proposed Work

A. *String Representation of Web Socuments:*

Based on the string representation of trees as proposed by M. J. Zaki [2], an algorithm has been proposed that will read each web page, which is basically assumed to be a HTML document, and will convert the same to a string.

The Algorithm is as follows-

Procedure preorder_depth_first (node node)

```
1. {
2.    if (node !=NULL)
3.       {
4.          print "[node_name]"
             // Print name of the HTML tag
5.          print ","
// Print a separator "," between two HTML tags
6.          for_eachchildnode c of node
7.           {
8.          pre_order_depth_first(c)
                 //  Recursively call the procedure on each
                  // children of the parent
9.               print "$"
                  // Print "$" to backtrack from child to parent
10.         }          // end_for
11.      } // end_if
12.  }     // end of Procedure
```

As shown in the above algorithm, the algorithm uses a recursive call to the procedure pre_order_depth_first() to traverse all nodes in the HTML DOM-Tree in a depth-first manner and prints the node names, separated by comma(,). Whenever the algorithm reaches a child node, it backtracks to the parent node of that child node and marks the backtracking by a ($). Although pre-order traversal of a tree is not sufficient to get the tree back, as different trees may have the same pre-order traversal, the output produced by the above algorithm is unique for each tree, and the same input tree can be constructed back from the output of the algorithm.

The input and output is shown below:

Input:

```
<HTML>
   |----<HEAD>
   |       |----<TITLE>MY PAGE</TITLE>
   |      |----<SCRIPT></SCRIPT>
   |   </HEAD>
   |----<BODY>
          |---- <DIV>
                  |---- <B>THIS IS
                           |---- <I>MY
                                    |----<U>PAPER</U>
                          </I>
</B>
</DIV>
</BODY>
</HTML>
```

Output:

html,body,p,$title,$script,$div,b,i,u,$$$$$

The prosed work aims at applying various distance measures one-by-one to the available string representation of a dataset of 314 web pages, and compare the time taken for calculation of string distances between each string i.e. each document.

Since time taken for the entire matric (314x314) was too long, the experiment has been carried out for first 120 documents.

V. EXPERIMENTAL RESULTS

A. *Implementation Details:*

- ***Hardware Environment:*** Intel Core2 Duo CPU T6570 2.10 GHz, 3.00 GB DDR2 RAM.
- ***Software Environment:*** Windows-7 32 bit, R version 3.2.2, RStudio Version 0.99.489.

B. *Description of Dataset used (UW-CAN-DATASET):*

A collection of web documents used for web mining purposes. The document data set has been used for testing the effect of phrase-based document similarity calculation, ascompared to using traditional single-term similarity measures. The dataset was primarily used for the work presented in [4]. The data set consists of 314 web pages from various websites at the University of Waterloo, and some Canadian websites. The data is categorized into 10 categories. Each category is in a separate folder.

URL:http://pami.uwaterloo.ca/~hammouda/webdata/uw-can-data.zip

(Download size: 1.14 MB; after Unzip: 4.12 MB)

C. *Experimental Results obtained:*

TABLE I. SUMMARY OF TIME TAKEN

Method	Time Taken (in Seconds)				
	n=10	**n=20**	**n=50**	**n=100**	**n=120**
OSA	0.4690001	2.615219	10.28083	48.5179	91.73538
LV	0.383415	2.073407	7.961015	40.35407	77.12256
DL	0.8268011	4.438214	20.59144	94.57536	136.68576
LCS	0.1872001	1.312402	5.042809	27.73125	48.10329
QGRAM	0	0.0155999	0.06239986	0.2496011	0.312
COSINE	0.0155999	0.0155999	0.04680014	0.2184	0.3598011
JACCARD	0	0.0155999	0.0467999	0.234	0.2184
JW	0.0155999	0.0780010	0.3442011	1.794003	2.814006
n=number of strings					

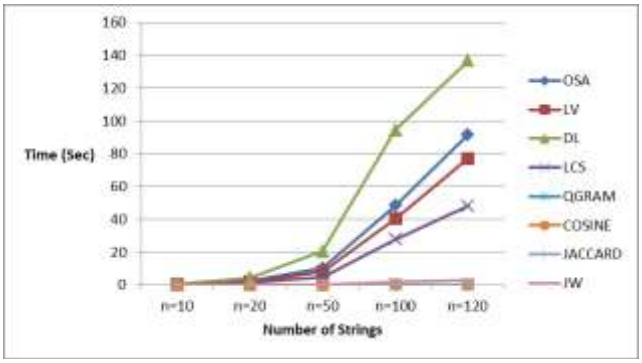

Figure 1. Comparison of Time Taken versus number of Documents

From the experimental results, it has been observed that the Jaccard distance has been found to be the least time consuming and the full DamerauLevensthein distance (DL) has been found to be the most time consuming among all the measures.

VI. CONCLUSION AND FUTURE WORKS

The work presented here aims to find out an optimal distance measure that can be used for clustering of web documents. The work further aims at applying these distances for efficient clustering of web documents where clustering will be performed based on not only the web page content but also the structural layout of a web page.

REFERENCES

[1] Jaideep Srivastava, Prasanna Desikan and Vipin Kumar, "Chapter 3: Web Mining- Accomplishments & Future Directions", URL: www.ieee.org.ar/downloads/Srivastava-tut-paper.pdf.

[2] G. Navarro (2001). A guided tour to approximate string matching. ACM Computing Surveys 33, 2001. p.p 31-88.

[3] Mohammed J. Zaki, "Efficiently Mining Frequent Trees in a Forest", Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, July 2002.

[4] Hammouda, K. and Kamel, M. "Phrase-based Document Similarity Based on an Index Graph Model", In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi, Japan, 2002. IEEE Computer Society.

[5] Rui Xu, Donald C. Wunsch II, "Survey of Clustering Algorithms", IEEE, IEEE Transactions on Neural Networks, VOL. 16, NO. 3, May 2005.

[6] Pradnya Purandare, "Web Mining: A Key To Improve Business On Web", IADIS European Conference Data Mining 2008. ISBN: 978-972-8924-63-8 © 2008 IADIS.

[7] Kabita Thaoroijam, "Document Clustering Using a Fuzzy Representation of Clusters", Ph. D. Thesis submitted to the Department of Computer Science, Gauhati University, Guwahati. 2011.

[8] L. Boytsov, "Indexing methods for approximate dictionary searching: comparative analyses. ACM Journal of experimental algorithmics, 16, 2011. p.p. 1-88.

[9] Manoj Pandia, Subhendu Kumar Pani, Sanjay Kumar Padhi, Lingaraj Panigrahy, and R. Ramakrishna, "A Review Of Trends In Research On Web Mining", International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1, 2011.

[10] Chhavi Rana, "Trends in Web Mining for Personalization", IJCST Vol. 3, Issue 1, Jan-March 2012. ISSN: 0976-8491 (Online), ISSN: 2229-4333 (Print).

[11] A. B. Manwar, Hemant S. Mahalle, K. D. Chinchkhede, and Vinay Chavan, "A Vector Space Model For Information Retrieval: A Matlab Approach", Indian Journal of Computer Science and Engineering (IJCSE) Vol. 3 No. 2 Apr-May 2012, ISSN : 0976-5166, pp. 222-229.

[12] https://www.w3.org/TR/WD-DOM/introduction.html.

[13] MPJ van der Loo, "The stringdist package for approximate string matching", The R Journal 6(1).2014. p.p. 111-122.