ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

A Supervised Technique for Web Document Detection, Extraction and HTML Noise Removal

Grace Mary Kurian,Shiju Shaikh Manakkulam

M.Tech Student, Dept. of C.S.E, ICET, Muvattupuzha, Ernakulam, Kerala,India

HOD, Dept. of M.C.A, ICET, Muvattupuzha, Ernakulam, Kerala, India

ABSTRACT: Internet is the collection of web pages in different formats.Web document detection and extraction causes many problems nowadays.Noise in the web documents create additional storage space and time complexity.It also degrades the efficiency of web document detection and extraction.In this paper,we present a new approach of how to detect and extract similar web documents and URL's with noise removal. Our approach is a supervised proposal that takes documents from the same server-side template and learns a regular expression from it and it can be used to detect and extract similar documents from the web.It uses the concept of DOM trees,so that the noise can be removed from the leaf nodes.We compared our technique to others on the large collection of web documents,it performs better than the others.Our results depicts that our proposal performs better than the others in memory usage,time complexityand effect of threshold because of noise removal. Nodes in Mobile Ad Hoc Networks (MANETs) are limited battery powered. That's why energy efficient routing has become an important optimization criterion in MANETs. The conventional routing protocols do not consider energy of the nodes while selecting routes which leads to early exhaustion of nodes and partitioning of the network. This paper attempts to provide an energy aware routing algorithm. The proposed algorithm finds the transmission energy between the nodes relative to the distance and the performance of the algorithm is analyzed between two metrics Total Transmission energy of a route and Maximum Number of Hops. The proposed algorithm shows efficient energy utilization and increased network lifetime with total transmission energy metric.

KEYWORDS: Document extraction, expression tree, noise displacement, supervised learning, web document detection

I. INTRODUCTION

The World Wide Web's vast growth and popularity has resulted in countless information sources on the internet.The information is represented in different user friendly formats.This does not yield common layout for all types of dataIt poses various challenges in extracting and detecting web documents of similaritywith less noise. The problem of translating the content of input documents into structured data is called information extraction.It concerns how to identify relevant documents from a document collection, IE produces structured data ready for post-processing, which is crucial to many applications of Web mining and searching tools.An IE task is defined by its input and its extraction target. The input can be unstructured documents like free text that are written in natural language or the semi-structured documents. Programs that perform the task of IE are referred to as extractors or wrappers. A wrapper was originally defined as a component in an information integration system which aims at providing a single uniform query interface to access multiple information sources. The amount of information that is currently available on the net in HTML format grows at a very fast pace,so that we may consider theWeb as the largest "knowledge base" ever developed and made available to the public. However HTML sites are in some sense modern legacy systems, since such a large body of data cannot be easily accessed and manipulated. The reason is that Web data sources are intended to be browsed by humans, and not computed over by applications. As a consequence, extracting data from Web pages and making it available to computer applications remains a complex and relevant task.Web Document detection is the process of finding out documents with similar structure and content.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

In this article, we introduce a technique web document detection with noise removal,which is an unsupervised proposal that learns extraction rules from a set of web documents that were generated by the same server-side template. It builds on the hypothesis that shared patterns are not likely to provide any relevant data and are, thus, part of the template. Whenever it finds a shared pattern, it partitions the input documents into the prefixes, separators and suffixes that they induce and analyses the results recursively, until no more shared patterns are found. Prefixes, separators, and suffixes are organised into a trinary tree that is later traversed to build a regular expression with capturing groups that represents the template that was used to generate the input documents.Web document detection with noise removal is a supervised technique to detect the similar documents.

The rest of the article is organised as follows:Section II presents the Literature survey. Section III presents the algorithms of web document detection;Section IV analyses on our experimental results; Section V concludes our work

II. RELATED WORK

The three most closely-related proposals are Road-Runner ExAlg, and FiVaTech [4], which differ significantly from Trinity: RoadRunner[2] is a parsing-based approach that uses a partial rule (which is initialised to any of the input documents) to parse another document and applies a number of generalisation strategies to correct the partial rule when mismatches are found; ExAlg finds maximal classes of tokens that occur in every input document, which are thus very likely to belong to the template, and then refines them using a token differentiation and a nesting criteria in order to construct the extraction rule; FiVaTech first identifies nodes in the input DOM trees that have a similar structure and then aligns their children and mines repetitive and optional patterns to create the extraction rule. Trinity,which is an unsupervised proposal that learns extraction rules from a set of web documents that were generated by the same server-side template.It is an unsupervised proposal that contains noise in the html documents.This increases the memory complexity,time complexity and effect of threshold.It only considers the first few attributes of the first record and last few attributes of the last record differently from the remaining records..So,it does not provide web document detection of similar documents accurately.

To overcome this limitation,we propose a supervised technique of web data document detection with noise removal by less memory complexity and time complexity.It helps us to detect similar web documents and URL's with noise removal.

III. PROPOSED ALGORITHM

In this section, we first provide an intuitive introduction to our proposal, finally provide an algorithmic description.The algorithm first creates a root node with the input web documents and sets a variable called *s* to *max*. Starting with this node, the algorithm loops and searches for a shared pattern of size *s*. If such a pattern is found in the current node, then it is used to create three new child nodes with the prefixes, the separators, and the suffixes that this pattern induces; the prefixes are the fragments from the beginning of a Text up to the first occurrence of a shared pattern, the separators are the fragments in between successive occurrences, and the suffixes are the fragments from the last occurrence until the end of a Text. These nodes are analysed recursively in order to find new shared patterns that induce new nodes. If no shared pattern is found, that is, the tree is not expanded, but variable *s* is greater or equal to the minimum pattern size, then *s* is decreased and the procedure is repeated again until a node in which no shared pattern of size greater or equal to *min* is found.

Fig. 1 shows a sample trinary tree. Node *N*1 represents three sample input documents; the algorithm searches for the longest shared pattern, which is underlined, and then creates three new nodes with the prefixes, the separators,and the suffixes that it induces. Note that the shared pattern is found at the beginning of the input documents, so the prefixes are empty strings that we represent using symbol ε; note that the shared pattern occurs only once, which implies that there are not actually any separators, which we represent with symbol *nil*. The processing continues recursively on the new nodes. Note that leaf nodes that have variability, that is, contain different fragments, have data that is not likely at all to belong to the template. Once the trinary tree is built, we need to use an additional algorithm to learn the regular expression that represents the template used to generate the input web documents.

This algorithm traverses the trinary tree in preorder; every time it reaches a leaf node that has variability, it outputs a fresh capturing group to extract the data that corresponds to that node; otherwise, it outputs the shared pattern that

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

corresponds to the node being analysed and a closure or an optional operator depending on whether that node is repeatable (there can be multiple occurrences of the shared pattern) or optional.

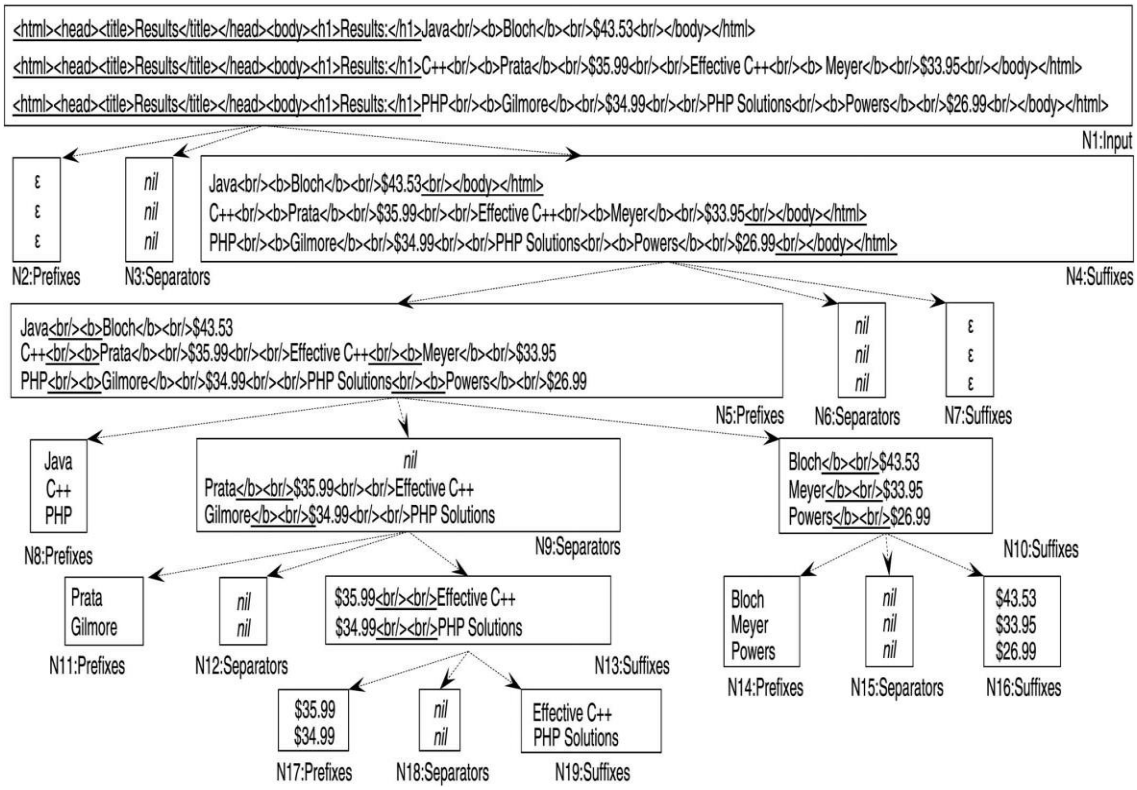

Fig 1.Sample Trinary Tree

```
<html><head><title>Results</title></head><body> <h1>Results:</h1>
{A}  <br/><b> ( ( {B} </b><br/> {C} <br/><br/> {D} )? <br/><b> )*
{E} </b><br/> {F}  <br/></body></html>
```

(a)

```
<html><head><title>Results</title></head><body> <h1>Results:</h1>
( {title} (<br/><b>)+ {author} </b><br/> {price} )+<br/><br/>
<br/></body></html>
```

(b)

Fig 2.Regular expression learnt from sample trinary tree.(a)Before Simplification.(b)After Simplification

Fig. 2(a) shows the regular expression that Trinity learns for the sample trinary tree in Fig. 1; capturing groups are represented as {A}, {B}, . . . , {F}.But here incudes noise in the web documents.To remove this and detect similar web documents we want to use the noise displacement algorithm.By using noise displacement algorithm we can remove unwanted and unnecessary lines from the html documents.Thus it increase the time of execution and it requires less amount of memory.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

1.detectDocument(node:Node;map:Map<Text,list<nat>>; pattern:Text)
2.prefixes=newNode()
3.separators=newNode()
4.suffixes=newNode()
5.set pattern of node to pattern
6.for each text in node **do**
7.removeNoise(children)
8.matches=map(noiseremovedText)
9.addComputePrefix(matches,noiseremovedText)to prefix
10.addComputeSeparators(matches,noiseremovedText)to separators
11.addComputeSuffix(matches,noiseremovedText)to suffixes
12.end
13.set prefix of node to prefixes
14.set separator of node to separators
15.set suffix of node to suffixes
16.end

Fig 3.Algorithm for detectDocument

1.removeNoise(node:Node;root:Node;leaf:Node<Text,list
<nat>>document:Text)
2.set root=maxTagValue
3.set leaf=minValue
4.remove node=(leaf1+leaf2)/2;
5.compute the removedNoiseDocument

Fig 4.Algorithm for removeNoise

Here the similar documents and URL's are detected with the help of noise displacement algorithm and DOM trees.In a document give the root tag maximum value and give less values by coming to the leaf node.Mainly the leaf nodes contain noise.If noise resides in the document,we need more memory and time to process it.It increases the creation time also.To decrease this noise from the web documents that we detect and extract use the Noise Displacement algorithm.To get the new position weight and new tag weight divide the position weight and tag weight by constants of each one.Adding new node's position weight and new node's tag weight gives the new node's total weight.Total weight of the new node is got by dividing the sum of node's total weight and childweight by three.

IV. SIMULATION RESULTS

In this section, we describe our experimental results, and finally analyse them.

Memory Complexity

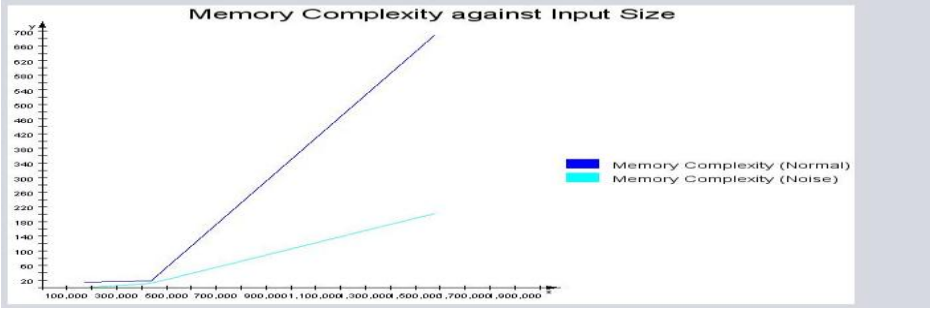

Fig 5. Comparison of Memory Complexity with and without noise removal

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

Figure 5 shows the comparison of memory complexity with and without noise removal.The graph shows that at small range of input size memory complexity is almost same for documents with and without noise removal .But the documents with noise removal has less memory complexity.The range of input documents is more,the memory complexity of documents with and without noise removal shows vast change in memory complexity.Web documents with noise removal utilizes less memory for storage.

Time complexity

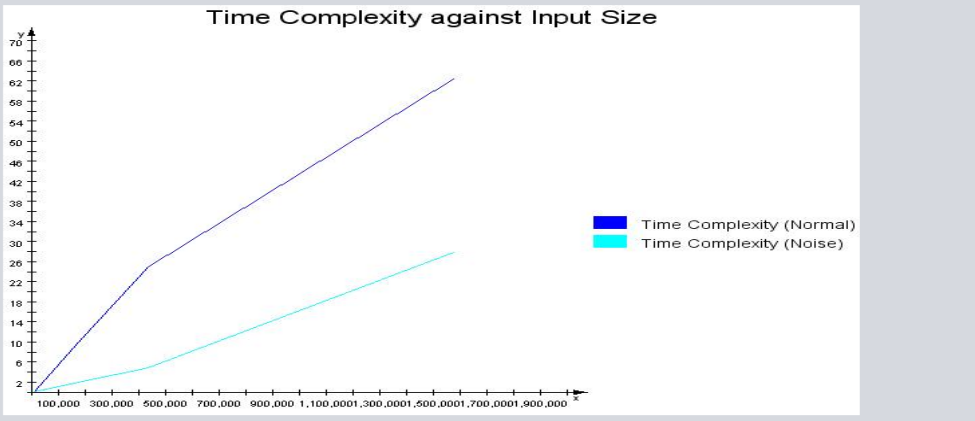

Fig 6. Comparison of Time Complexity with and without Noise Removal

Figure 6 shows the comparison of time complexity with and without noise removal.The graph shows that in all range of input size time complexity varies from starting itself.Web documents are fastly processed with noise removal algorithm.So,it increases the whole execution process.

Effect of Detection Rate

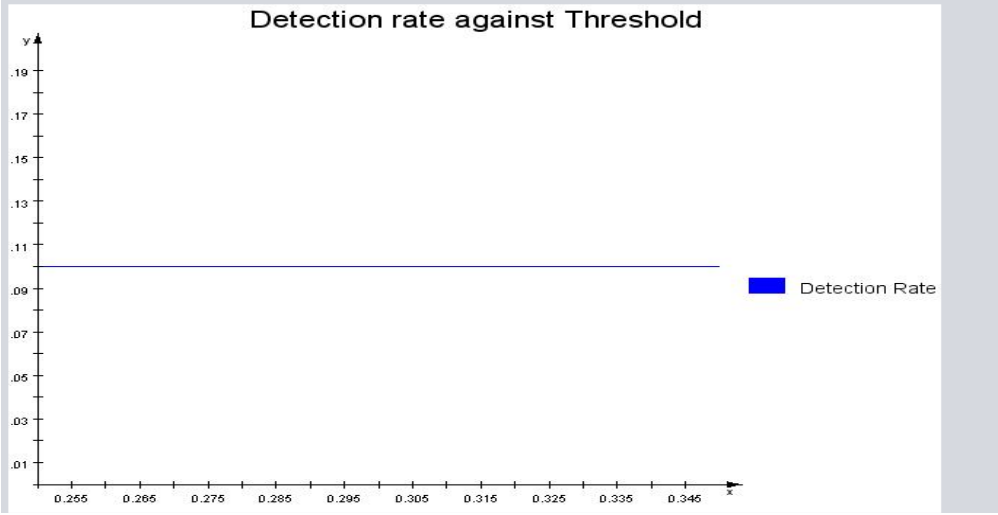

Fig 7.Detection rate against threshold

Figure 7 shows the detection rate of documents against threshold.It shows uniform detection of documents with and without noise removal.It means that all documents are processed at the same detection rate.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

V. CONCLUSION AND FUTURE WORK

We have presented an effective supervised data extractor and detector with noise removal.. It is based on the hypothesis that web documents generated by the same server-side template,share patterns that do not provide any relevant data, but help delimit them. The noise displacement algorithm searches for these patterns and creates a trinary tree, which is then used to learn a regular expression that represents the template that was used to generate input web documents.Our experiments proved that our technique achieves less memory to store,very high speed and uniform detection rate.It provides detection of web documents and URL's with high speed and less memory storage space.

REFERENCES

1. V. Crescenzi and G. Mecca, "Automatic information extraction from large websites,"*J. ACM*, vol. 51, no. 5, pp. 731–779, Sept. 2004
2. V. Crescenzi, G. Mecca, and P. Merialdo, "Road runner: Towards automatic data extraction from large web sites," in *Proc. 27th Int. Conf. VLDB*, Rome, Italy, 2001, pp. 109–118.
3. A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proc. 2003 ACM SIGMOD*, San Diego, CA, USA,pp. 337–348.
4. M. Kayed and C.-H. Chang, "FiVaTech: Page-level web data extraction from template pages," *IEEE Trans. Knowl. Data Eng.*,vol. 22, no. 2, pp. 249–263, Feb. 2010.
5. C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1411–1428, Oct. 2006.
6. W. W. Cohen, M. Hurst, and L. S. Jensen, "A flexible learning system for wrapping tables and lists in HTML documents," in *Proc. 11th Int. Conf. WWW*, 2002, pp. 232–241.
7. H. Elmeleegy, J. Madhavan, and A. Y. Halevy, "Harvesting relational tables from lists on the web," in *Proc. VLDB*, vol. 2, no. 1, pp. 1078–1089, Aug. 2009
8. P. Gulhane, R. Rastogi, S. H. Sengamedu, and A. Tengli, "Exploiting content redundancy for web information extraction," in *Proc. 19th Int. Conf. WWW*, Raleigh, NC, USA, 2010,pp. 1105–1106.
9. P. Gulhane *et al.*, "Web-scale information extraction with vertex," in *IEEE 27 ICDE*, Hannover, Germany, 2011, pp. 1209–1220
10. R. Gupta and S. Sarawagi, "Answering table augmentation queries from unstructured lists on the web," in *Proc. VLDB*, vol. 2,no. 1, pp. 289–300, Aug. 2009

BIOGRAPHY

Grace Mary Kurian is an M.Tech student in the Computer Science and Engineering Department, Ilahia College Of Engineering And Technology, Muvattupuzha, Ernakulam,Kerala,India.Her research interests are Data Mining, Privacy Preserving Data Mining, Web Mining etc.

Shiju Shaikh Manakkulam is the Head of the Department of Master of Computer Applications,Ilahia College of Engineering and Technology, Muvattupuzha, Ernakulam, Kerala, India.His research interests are Privacy Preserving Data Mining,Web Mining, Security, Cryptography etc.