Template Extraction from Heterogeneous Web Pages Using Text Clustering

T.L.N.Divya[1], G.Loshma2, Dr. Nagaratna P Hegde[3]

[1] M.Tech(CSE), Sri Vasavi Engineering College, Pedatadepalli,Tadepalligudem
[2] Associate professor(CSE), Sri Vasavi Engineering College, Pedatadepalli,Tadepalligudem
[3] Associate professor(CSE), Vasavi College of Engineering, Hyderabad

***Abstract* - Now a days most of the information is stored in text databases. This information consists of large collection of documents from Heterogeneous web pages. Now we extract template from these heterogeneous templates, and to extract template we use different algorithms to find similarity of underlying template structures in the documents and we cluster the web documents based on the similarity of underlying template structure in the documents so that template is extracted with various clusters. We use different algorithms to find similarity between the web pages. Previously the algorithms used are RTDM, Text-Hash and Text-Max. But the time and space occupied by this algorithms is more. In this paper we are using WaveK-Means algorithm to find similarity between the web pages. This algorithm provides better performance compared to previous algorithms in terms of space and time. The space and time consumed by this algorithm is less compared to RTDM, Text-Hash and Text-Max. Our Experimental results with real life data sets confirm effectiveness and robustness of our algorithm.**

***Keywords* - Template Extraction, RTDM, Text-Hash, Text-Max, WaveK-means, Clustering.**

I. INTRODUCTION

World Wide Web is the most useful source of information. In order to achieve high productivity of publishing, the WebPages in many websites are automatically populated by using the common templates with contents. The templates provide readers easy access to the contents guided by consistent structures. However, for machines, the templates are considered harmful since they degrade the accuracy and performance of web applications due to irrelevant terms in templates. Thus, template detection techniques have received a lot of attention recently to improve the performance of search engines, clustering, and classification of web documents.

The Web poses itself as the largest data repository ever available in the history of humankind[1]. Major efforts have been made in order to provide efficient access to relevant information within this huge repository of data. Although several techniques have been developed to the problem of Web data extraction, their use is still not spread, mostly because of the need for high human intervention and the low quality of the extraction results. In this paper, we present a domain-oriented approach to Web data extraction and discuss its application to automatically extracting news from Web sites. Our approach is based on a highly efficient tree structure analysis that produces very effective results. We have tested our approach with several important Brazilian on-line news sites and achieved very precise results, correctly extracting 87.71% of the news in a set of 4088 pages distributed among 35 different sites.

In this paper, we present novel algorithms for extracting templates from a large number of web documents which are generated from heterogeneous templates. We cluster the web documents based on the similarity of underlying template structures in the documents so that the template for each cluster is extracted simultaneously. We develop a novel goodness measure with its fast approximation for clustering and provide comprehensive analysis of our algorithm. Our experimental results with real-life data sets confirm the effectiveness and robustness of our algorithm compared to the state of the art for template detection algorithms.

II. RELATED WORK

Document Classification

One approach to solving the metadata extraction problem for a heterogeneous collection is to partition the collection into a set of homogeneous collections first and then solve the extraction problem for each homogeneous collection. Document classification is used to create equivalence groups of similar documents. Few researchers have addressed the problem of how to find the page(s) that will be used to differentiate the documents.

Existing approaches to classify documents (assuming that one has the page containing the metadata isolated) into equivalence groups include one that uses a document model based upon page layout structure.

The goal of the DOM specification is to define a programmatic interface for XML and HTML[2]. The DOM Level 1 specification is separated into two parts: Core and HTML. The Core DOM Level 1 section provides a low-level set of fundamental interfaces that can represent any structured document, as well as defining extended interfaces for representing an XML document. These extended XML interfaces need not be implemented by a DOM implementation that only provides access to HTML documents; all of the fundamental interfaces in the Core

section must be implemented. A compliant DOM implementation that implements the extended XML interfaces is required to also implement the fundamental Core interfaces, but not the HTML interfaces. The HTML Level 1 section provides additional, higher-level interfaces that are used with the fundamental interfaces defined in the Core Level 1 section to provide a more convenient view of an HTML document. A compliant implementation of the HTML DOM implements all of the fundamental Core interfaces as well as the HTML interfaces.

This section extends the Level 1 Core API to describe objects and methods specific to HTML documents. In general, the functionality needed to manipulate hierarchical document structures, elements, and attributes will be found in the core section; functionality that depends on the specific elements defined in HTML will be found in this section.

The goals of the HTML-specific DOM API are:

- to specialize and add functionality that relates specifically to HTML documents and elements.
- to address issues of backwards compatibility with the "DOM Level 0".
- to provide convenience mechanisms, where appropriate, for common and frequent operations on HTML documents.

The term "DOM Level 0" refers to a mix (not formally specified) of HTML document functionalities offered by Netscape Navigator version 3.0 and Microsoft Internet Explorer version 3.0. In some cases, attributes or methods have been included for reasons of backward compatibility with "DOM Level 0".

The key differences between the core DOM and the HTML application of DOM is that the HTML Document Object Model exposes a number of convenience methods and properties that are consistent with the existing models and are more appropriate to script writers. In many cases, these enhancements are not applicable to a general DOM because they rely on the presence of a predefined DTD. For DOM Level 1, the transitional and frameset DTDs for HTML 4.0 are assumed. Interoperability between implementations is only guaranteed for elements and attributes that are specified in these DTDs.

More specifically, this document includes the following specializations for HTML:

- An HTML Document interface, derived from the core Document interface. HTML Document specifies the operations and queries that can be made on a HTML document.
- An HTML Element interface, derived from the core Element interface. HTML Element specifies the operations and queries that can be made on any HTML element. Methods on HTML Element include those that allow for the retrieval and modification of attributes that apply to all HTML elements.
- Specializations for all HTML elements that have attributes that extend beyond those specified in the HTML Element interface. For all such attributes, the derived interface for the element contains explicit methods for setting and getting the values.

The DOM Level 1 does not include mechanisms to access and modify style specified through CSS 1. Furthermore, it does not define an event model for HTML documents. This functionality is planned to be specified in a future Level of this specification.

HTML Application of Core DOM

Naming Conventions

The HTML DOM follows a naming convention for properties, methods, events, collections, and data types. All names are defined as one or more English words concatenated together to form a single string. Properties and Methods

The property or method name starts with the initial keyword in lowercase, and each subsequent word starts with a capital letter. For example, a property that returns document meta information such as the date the file was created might be named "fileDateCreated". In the ECMA Script binding, properties are exposed as properties of a given object. In Java, properties are exposed with get and set methods. Non-HTML 4.0 interfaces and attributes

While most of the interfaces defined below can be mapped directly to elements defined in the HTML 4.0 Recommendation, some of them cannot. Similarly, not all attributes listed below have counterparts in the HTML 4.0 specification (and some do, but have been renamed to avoid conflicts with scripting languages). Interfaces and attribute definitions that have links to the HTML 4.0 specification have corresponding element and attribute definitions there; all others are added by this specification, either for convenience or backwards compatibility with "DOM Level 0" implementations.

Rule-based Approach

The steps of building a rule-based metadata extraction system are typically as follows: first, some experts examine samples of the document collection and define rules for metadata extraction; then, software developers implement these rules

either as part of an expert system or as part of an ad hoc rule engine. The accuracy, inventiveness, and appropriateness of the rules that experts defined play a critical role in building a system with high accuracy.

III. PROPOSED APPROACH

List of Modules:

1. DOM and Initial approaches
2. Clustering with MDL cost
3. Optimal template path calculation for clusters
4. MDL cost estimation using Min Hash
5. Clustering with Min Hash

DOM and Initial Approaches:

This module describes about the Document Object Model (DOM) and initial approaches, they are in the following,

DOM: The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page.

In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions - in particular, the DOM interfaces for the XML internal and external subsets have not yet been specified.

The DOM defines a standard for accessing documents, like HTML and XML. The DOM presents an HTML document as a tree structure. The entire document is a document node, every HTML element is an element node, the texts in the HTML elements are text nodes, every HTML attribute is an attribute node, and comments are comment nodes. However, we do not distinguish the type of nodes, since, as defined, any type of node can be a part of a template in our problem. For instance, the DOM tree of a simple HTML document d2 in Fig. 2b is given in Fig. 1. For a node in a DOM tree, we denote the path of the node by listing nodes from the root to the node in which we use "n" as a delimiter between nodes. For example, in the DOM tree of d2 in Fig. 1, the path of a node "World" is "Document\<html>\<body> \< h1>\ in World."

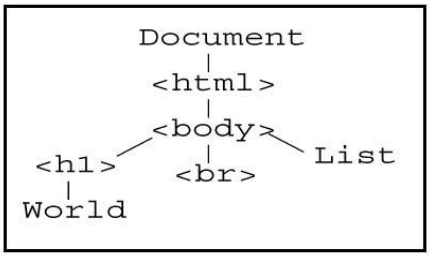


Fig : 1. DOM Tree of Document d2

Essential Paths and Templates: Given a web document collection D, we define a path set Pd as the set of all paths in D. Note that, since the document node is a virtual node shared by every document, we do not consider the path of the document node in Pd. The support of a path is defined as the number of documents in D, which contain the path. For each document di, we provide a minimum support threshold tdi . Notice that the thresholds tdi and tdj of two distinct documents di and dj, respectively, may be different. If a path is contained by a document di and the support of the path is at least the given minimum support threshold tdi , the path is called an essential path of di.

Matrix Representation of Clustering: An approximation or representation to a matrix A can be thought of as a system which captures the most "important" information in A. Obtaining the representation usually involves a tradeoff among accuracy, the memory space occupied by the representation, and the time required to obtain the representation. These three aspects of the representation are usually in conflict, and must be tailored to fit the desired application. Unlike the recent low-rank representation methods , this representation is not used to extract directly the essential concepts that pervade an entire dataset, in which each new "concept vector" is forced to be independent (orthogonal in some cases) of the preceding "concept vectors[3]."

We next illustrate the representation of a clustering of web documents. Let us assume that we have m clusters for a web document set D. A cluster ci , Ti is a set of paths representing the template of ci and Di is a set of documents belonging to ci. In our clustering model, we allow a document to be included in a single cluster only. To represent a clustering information for D, we use a pair of matrices MT and MD, where MT represents the information of each cluster with its template paths and MD denotes the information of

each cluster with its member documents. We will represent ME by the product of MT and MD. However, the product of MT and MD does not always become ME.

Minimum Description Length Principle: In order to manage the unknown number of clusters and to select good partitioning from all possible partitions of HTML documents, we employ Rissanen's MDL principle. The MDL principle states that the best model inferred from a given set of data is the one which minimizes the sum of

1) the length of the model, in bits, and
2) the length of encoding of the data, in bits, when described with the help of the model.

We refer to the above sum for a model as the MDL cost of the model. In our setting, the model is a clustering C, which is described by partitions of documents with their template paths (i.e., the matrices MT and MD), and the encoding of data is the matrix M.

Clustering with MDL cost:

This module describes the clustering of documents using MDL cost[4]. The input parameter is a set of documents D, where di is the ith document. The output result is a set of clusters C, where ci is a cluster represented by the template paths Ti and the member documents Di. A clustering model C is denoted by two matrices MT and MD and the goodness measure of the clustering C is the MDL cost, which is the sum of TEXT-MDL is an agglomerative hierarchical clustering algorithm which starts with each input document as an individual cluster. When a pair of clusters is merged, the MDL cost of the clustering model can be reduced or increased. The procedure GetBestPair finds a pair of clusters whose reduction of the MDL cost is maximal in each step of merging and the pair is repeatedly merged until any reduction is not possible. In order to calculate the MDL cost when each possible pair of clusters is merged, the procedure GetMDLCost(ci, cj, C), where ci and cj are a pair to be merged and C is the current clustering, is called in GetBestPair and C is updated by merging the best pair of clusters. As we will discuss later in detail, because the scale of the MDL cost reduction by merging a pair of clusters is affected by all the other clusters, GetBestPair should recalculate the MDL cost reduction of every pair at each iteration. Furthermore, the complexity of GetMDLCost is exponential on the size of the template of a cluster. Since it is not practical to use TEXT-MDL with a number of web documents, we will introduce an approximate MDL cost model and use MinHash to significantly reduce the time complexity.

MDL cost estimation using MinHash:

This module describes the estimation of MDL cost using MinHash, that are given in the below lines. To compute the MDL cost of each clustering quickly, we would like to estimate the probability that a path appears in a certain number of documents in a cluster. However, the traditional MinHash was proposed to estimate the Jaccard's coefficient. Thus, given a collection of sets X= (S1, ... Sk), we extend MinHash to estimate the probabilities needed to compute the MDL cost. Recall that, if we know SK;D for each pk, we can decide the optimal Ti and calculate the MDL cost of a clustering. Note that we estimate the MDL cost, but do not generate the template paths of each cluster. Thus, Tk of ck is initialized as the empty set. Instead of the template paths, the signature of ck is maintained to estimate the MDL cost. If we consider the length of a signature as a constant, the complexity of GetHashMDLCost. After finishing clustering, a post processing is needed to get the actual template paths[5]. We refer to the processing as the template path generation step.

Clustering with MinHash:

This module describes the clustering with MinHash, we merge clusters hierarchically, we select two clusters which maximize the reduction of the MDL cost by merging them. Given a cluster ci, if a cluster cj maximizes the reduction of the MDL cost, we call cj the nearest cluster of ci. By using Heuristic 1, we can reduce the search space to find the nearest cluster of a cluster ci. The previous search space to find the nearest cluster of ci was the same as the number of current clusters. But, using Heuristic 1, the search space becomes the number of clusters whose Jaccard's coefficient with ci is maximal. The Jaccard's coefficient can be estimated with the signatures of MinHash and clusters whose Jaccard's coefficient with ci is maximal can be directly accessed in the signature space. We provide the procedures to find the best pair using MinHash. In GetInitBestPair[6], we first merge clusters with the same signature of MinHash. Next, for each cluster ci, we get clusters with the maximal Jaccard's coefficient estimated by the signatures of MinHash and compute the MDL cost of each pair. In GetHashBestPair, the steps are similar to those in GetInitBestPair. The complexities of GetInitBestPair and GetHashBestPair depend on the number of clusters with the maximal Jaccard's coefficient[7]t.

IV. EXPERIMENTAL RESULTS

All experiments were performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operation system platform is Microsoft Windows XP Professional (SP2)

Text clustering using Text-Hash

Cluster 1
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html

Cluster 2
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm

Cluster 3
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm

Cluster 4
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm
C:\Users\Aparna\Desktop\HTML\Test\Yahoo! India.htm
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm
C:\Users\Aparna\Desktop\HTML\Test\Yahoo! India.htm

Cluster 5
C:\Users\Aparna\Desktop\HTML\Test\Google.htm
C:\Users\Aparna\Desktop\HTML\Test\Google.htm

Cluster 6
C:\Users\Aparna\Desktop\HTML\Test\Home.htm

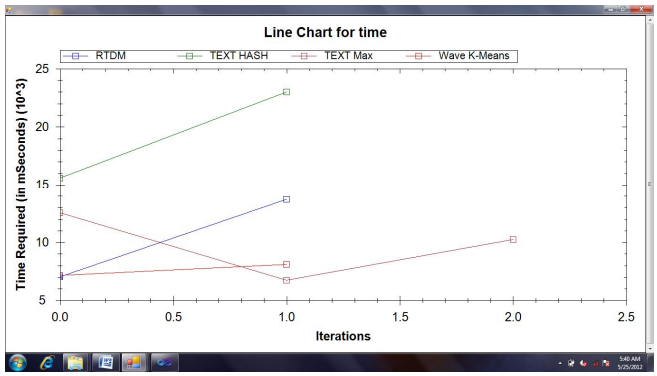
Text clustering using Text-Max

Cluster 1
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html

Cluster 2
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm

Cluster 3
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping

Line chart for time



Cluster 1
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html

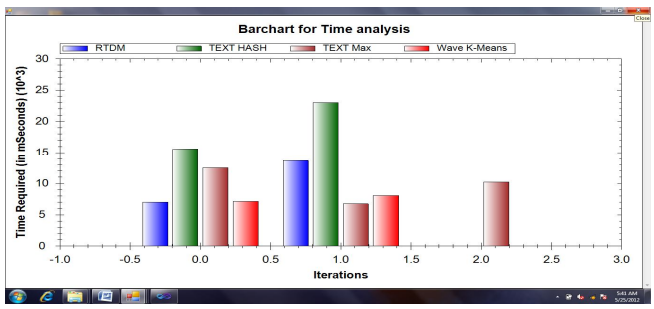
Cluster 2
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm

Cluster 3
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm

Cluster 4
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm
C:\Users\Aparna\Desktop\HTML\Test\Yahoo! India.htm
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm
C:\Users\Aparna\Desktop\HTML\Test\Yahoo! India.htm
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm
C:\Users\Aparna\Desktop\HTML\Test\Yahoo! India.htm

Text clustering using Wavek_Means

Bar chart for time



Cluster 1
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html
C:\Users\Aparna\Desktop\HTML\Test\0,28757,2087815,00.html

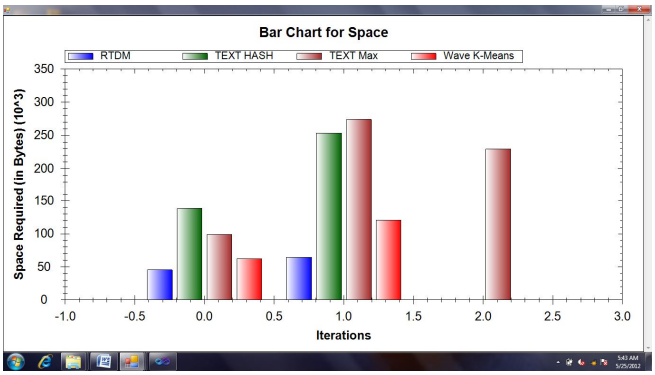
Cluster 2
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm
C:\Users\Aparna\Desktop\HTML\Test\Best Indian Websites.htm

Cluster 3
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm
C:\Users\Aparna\Desktop\HTML\Test\Buy Shoes Online in India for Men, Women & Kids at Shoe Shopping Store.htm

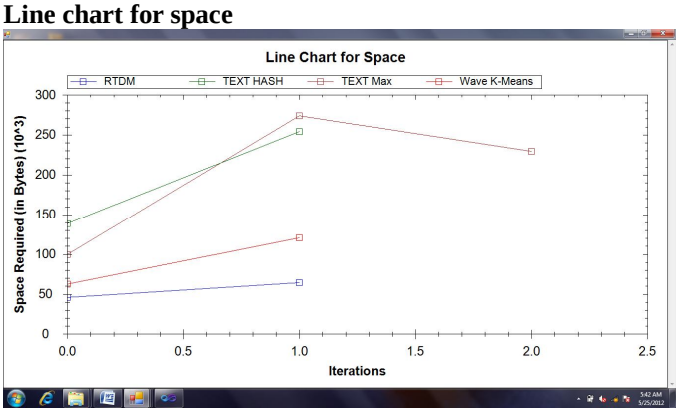
Cluster 4
C:\Users\Aparna\Desktop\HTML\Test\CineJosh - Telugu Movie Entertainment & Political Exclusive News, Photos, Videos, Spicy Gossips Website in English & Telugu Versions with Old Black & White Content.htm

Wave k-means

Bar chart for space



Line chart for space



5. CONCLUSION AND FUTURE WORK

This system executes less time usage for template extraction compare to existing algorithms like RTDM, Text-Hash and Text-Max. In this system we used WaveK-Means algorithm to find similarity between the web pages. This algorithm provides better performance compared to previous algorithms in terms of space and time. The space and time consumed by this algorithm is less compared to RTDM, Text-Hash and Text-Max. Our Experimental results with real life data sets confirm effectiveness and robustness of our algorithm.

Moreover, in future implementation our extraction approach will be resilient to changes in source document formats. For example, changes in HTML formatting codes do not affect our ability to extract and structure information from a given Web page. Finally the model will contribute effectively to the emergence of semantic web, by providing methodology, tools and both global and generic solutions.

REFERENCES

[1] Automatic web news extraction using tree edit distance by D C Reis, P B Golgher, A S Silva, A F Laender .

[2] http://www.w3.org/TR/REC-DOM-Level-1/.

[3] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. ACM SIGMOD, 2003.

[4] Z. Bar-Yossef and S. Rajagopalan, "Template Detection via Data Mining and Its Applications," Proc. 11th Int'l Conf. World Wide Web (WWW), 2002.

[5] A.Z. Broder, M. Charikar, A.M. Frieze, and M. Mitzenmacher, "Min-Wise Independent Permutations," J. Computer and System Sciences, vol. 60, no. 3, pp. 630-659, 2000.

[6] D. Chakrabarti, R. Kumar, and K. Punera, "Page-Level Template Detection via Isotonic Smoothing," Proc. 16th Int'l Conf. World Wide Web (WWW), 2007.

[7] Z. Chen, F. Korn, N. Koudas, and S. Muthukrishnan, "Selectivity Estimation for Boolean Queries," Proc. ACM SIGMOD-SIGACTSIGART Symp. Principles of Database Systems (PODS), 2000.