OPTIMIZED TEMPLATE BASED DETECTION MODEL FOR HETEROGENEOUS WEB PAGES

A PROJECT REPORT

Submitted by

S GAYATRI

in partial fulfillment for the requirement of award of the degree of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE 641 049
(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

BONAFIDE CERTIFICATE

Certified that this project work titled **"OPTIMIZED TEMPLATE BASED DETECTION MODEL FOR HETEROGENEOUS WEB PAGES"** is the bonafide work of S GAYATRI, (1120108005) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other students.

Prof. N.JAYAPATHI M.E.,	P. BETTY M.E (Ph.D).,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	**Assistant Professor**
Dept of Computer Science and Engineering	Dept of Computer Science and Engineering
Kumaraguru College of Technology	Kumaraguru College of Technology
Coimbatore- 641 049	Coimbatore -641049

Submitted for the Project Viva-Voce examination held on ____________.

--------------------------------	-------------------------------
Internal Examiner	**External Examiner**

ABSTRACT

World Wide Web is widely used to publish and access information on the Internet. The template detection techniques have been improved for providing a higher efficient access to the contents from the web pages. The approach of template detection is used to improve the performance of the search engines, clustering and classification of web documents. A novel algorithm called text automatic template extraction is proposed for extracting templates based on the similarity from large number of web documents. For machines, the unknown templates are considered harmful because they degrade the accuracy and performance due to the irrelevant terms in templates. Template detection aims to detect unimportant information of web pages and web collections that negatively impact the performance and resource usage of tools that process web pages. The proposed approach is enhanced by using a new similarity function named Locality Sensitive Hashing which is an algorithm for solving the approximate or exact Near Neighbour Search in high dimensional spaces. It starts with random projection operation that maps a data point from high dimensional to low dimensional subspace. It creates the projections from a number of different directions and keeps the track of nearby points. This method performs better than the existing method of text automatic template extraction.

ஆய்வுச்சுருக்கம்

உலகளாவிய வலைதள பரவல் என்பது தகவல்களை வெளியிடவும், அணுகவும் பயன்படுத்தப்படும். வார்புருக்களை கண்டறியும் நுட்பத்தை, மேம்படுத்தி, இணையதள தகவல்களை, திறம்பட அணுக முடியும். வார்புருக்களை கண்டறியும் அணுகுமுறை, இணையதள தேடுமுறையின் செயல்திறனை மேம்படுத்த, தொகுக்க மற்றும் இணையதள ஆவணங்களை வகைப்படுத்தவும் பயன்படுகிறது. ஒரு புதிய வழிமுறை அதாவது, தானாக வார்புருக்களை பிரித்தெடுக்கும் முறையை பயன்படுத்தி, ஒரே விதமான ஆவணங்களை இணையதளத்திலிருந்து பிரித்தெடுக்க முடிகிறது. தெரியாத வார்புருக்கள் என்பவை இயந்திரத்துக்கு தீங்கு விளைவிக்க கூடியவை. ஏனெனில் தேவையற்ற வார்த்தைகளை இவை கொண்டுள்ளது. இணையதளத்தில் உள்ள தேவையற்ற தகவல்களை கண்டறிவதே, வார்புருக்களை கண்டறியும் நுட்பத்தை நோக்கமாகும் உள்ளூர், முக்கிய வட்டார சுட்டு முகவரியாக்கம் என்கிற தொழில்நுட்பத்தை பயன்படுத்தி மேற்கண்ட முறையை விரிவுபடுத்துவதன் மூலம், ஏறக்குறைய ஒரே மாதிரியான தேடல்கள் பிரச்சனையை தீர்க்க முடியும். இவை முறைமையற்ற வெளிப்படுத்தல் என்பதிலிருந்து தொடங்குகிறது. இவை முக்கிய தகவல்களை கண்டறிந்து அவற்றை அடர்த்தி அதிகமான இடத்திலிருந்து அடர்த்தி குறைவான இடத்திற்கு மாற்றப்படுகிறது. இந்த வெளிப்படுத்தல் முறையானது அருகிலுள்ள தகவல்களை கண்டறிய பயன்படுகிறது. இந்த புதிய முறையின் செயல்திறன், பழைய முறையின் செயல்திறனை விட மேம்பட்டதாக உள்ளது.

ACKNOWLEDGEMENT

. First and foremost, I would like to thank the Lord Almighty for enabling me to complete this project.

I express my profound gratitude to **Padmabhusan Arutselvar Dr.N.Mahalingam, B.Sc., F.I.E , Chairman**, **Dr.B.K. Krishnaraj Vanavarayar**, **Co-Chairman, Mr. M. Balasubramaniam, M.Com., M.B.A, Correspondent, Mr.Sankar Vanavarayar**, **M.B.A., PGDIEM, Joint Correspondent** and **Dr.S.Ramachandran Ph.D., Principal** for providing the necessary facilities to complete my project.

I take this opportunity to thank **Prof.N.Jayapathi M.Tech**., Head of the Department, Department of Computer Science and Engineering, for his support and timely motivation. Special thanks to my Project Coordinator **Dr.V.Vanitha M.E., Ph.D.**, Senior Associate Professor, Department of Computer Science and Engineering, and project committee members for arranging project review sessions.

I register my sincere thanks to my Guide **Ms.P.Betty M.E.,(Ph.D)** Assistant Professor, Department of Computer Science and Engineering. I am grateful for her support, encouragement and ideas. I would like to convey my honest thanks to all **Teaching** and **Non Teaching** Staff members of the department and my classmates for their support.

I dedicate this project work to my **Parents** for no reasons but feeling from the bottom of my heart, without their love and sacrifice this work have not been possible.

S GAYATRI

LIST OF FIGURES

LIST OF ABBREVIATIONS

DOM	Document Object Model
DTD	Document Type Descriptor
HTML	Hyper Text Mark Up Language
MDL	Minimum Description Length
MSE	Multiple Section Extraction
RTDM	Restricted Top Down Mapping
SE	Search Engine
XML	Extensible Mark Up Language

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 WEB MINING

Web mining is the application of data mining techniques to discover patterns from the Web. According to analysis targets, web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining.

1.1.1 WEB USAGE MINING

Web usage mining is the process of extracting useful information from server logs users history. Web usage mining is the process of finding out what users are looking for on Internet

1.1.2 WEB STRUCTURE MINING

Web structure mining is the process of using graph theory to analyze the node a connection structure of a web site. According to the type of web structural data, web structure mining can be divided into two kinds:

1. Extracting patterns from hyperlinks in the web. A hyperlink is a structural component that connects the web page to a different location.

2. Mining the document structure analysis of the tree-like structure of page structures to describe HTML or XML tag usage.

1.1.3 WEB CONTENT MINING

Web content mining is the mining, extraction and integration of useful data, information and knowledge from Web page contents. The heterogeneity and the lack of structure that permits much of the ever expanding information sources on the World Wide Web, such as hypertext documents, makes automated discovery, organization, and search and indexing tools of the Internet and the World Wide Web such as Lycos, Alta Vista, WebCrawler, ALIWEB Meta-Crawler, and others provide some comfort to users, but they do not generally provide structural information nor categorize, filter, or interpret documents

1.2 WEB DATA EXTRACTION

Web Data Extraction is the process of pulling data from designated sites. Web data extraction can be performed by manual effort or by automated system.

.

It takes unstructured data from the web html pages and converts it to into structured records which are easy to access. In order to extract web data it is very essential to identify specific sites from which the search should be carried out.

1.3 DATA CLUSTERING

Data Clustering is a process of partitioning a set of data or objects in a set of meaningful subclasses, called clusters. Clustering helps users to understand the natural grouping or structure in a data set. Clustering is an unsupervised classification and it has no predefined classes. Cluster is a collection of data objects that are similar to one another called a group and they are sufficiently different from other groups. In clustering, the class labels and number of classes may not be known initially. The requirements of clustering methods are

- Scalability.
- Dealing with different types of attributes.
- Discovery of clusters with arbitrary shape.
- Minimal requirements for domain knowledge to determine input parameters.
- Able to deal with noise and outliers.
- Insensitive to order of input records.

1.4 WEBSITE

Websites like Amazon.com are data-intensive, and information on them comes from structured sources. Often the data are encoded into semi-structured HTML pages that employ templates for rendering. Some value-added services, such as comparison shopping, are emerging to query or manipulate the data, such as products and reviews, from several websites. To achieve high accuracy, the task of extracting structured information from Web pages is usually implemented by programs called wrappers. Manually writing wrappers for Web sources is a tedious, time-consuming, and error-prone job, thus the study of automatic wrapper induction using machine learning techniques has been a subject of research in recent years. A large number of

Web sites contain highly structured regions. The pages contained in these regions are generated automatically, either statically or dynamically, by programs that extract the data from a back-end database and embed them into an HTML template. As a consequence, pages generated by the same program exhibit common structure and layout, while differing in content. Based on this observation, several researchers have recently proposed techniques that leverage the structural similarities of pages from large Web sites to automatically derive Web wrappers, programs that extract data from HTML pages, and transform them into a machine accessible format, typically in XML. These techniques take a small set of sample pages that exhibit a common template, and generate a wrapper that can be used to extract the data from any page that shares the same structure of the input samples. Applying automatically generated wrappers on a large scale to the structured portion of the Web, could anticipate some of the benefits advocated by the Semantic Web vision, because large amounts of data exposed throughout HTML Web sites could become available for applications for example, the financial data published by several specialized Web sites only in HTML, could be constantly extracted and processed for mining purposes data delivered on the Web by thematic communities could be extracted and integrated.

1.5 WEB PAGE TEMPLATE

Templates can appear in primitive form, such as the default HTML code generated by HTML editors like Netscape Composer or FrontPage Express, or can be more elaborate in the case of large web sites. These templates sometimes contain extensive navigational bars that link to the central pages of the web site, advertisement banners, links to the FAQ and Help pages, and links to the web site's administrator page. The use of templates has grown with the recent developments in web site engineering. Many large web sites today are maintained automatically by programs, which generate pages using a small set of formats, based on fixed templates.

Templates can spread over many web sites and contain links to other web sites such as endorsement links to business partner web sites like www.eunet.net, advertisement links, and download links. Thus, traditional techniques to combat templates, like intra-site link filtering, are not effective for dealing with their new sophisticated form Since all pages that conform to a common template share many links, it is clear that these links cannot be relevant to the specific content on these pages. Thus templates violate both the Relevant Linkage Principle and the Topical Unity Principle. They may also cause violations of the Lexical Affinity Principle, if they are interleaved with the actual content of the pages. Therefore, improving hypertext data quality by recognizing and dealing with templates seems essential to the success of the hypertext IR tools.

A template has two characterizing properties:

1. There are a significant collection of pages that conform to this template.

2. This common look-and-feel of these pages are controlled or influenced by single property

The latter property is important in order to distinguish between templates, in which a collection of pages intentionally share common parts, and the following:

1. Mirrors complete wholesale duplication of pages or sites.

2. Independent pages that accidentally share similar parts, which might be important signatures of communities.

The requirement that a central authority influences all pages that conform to a template does not necessarily imply that they all belong to a single web site. The central authority can also be one that coordinates templates between sister sites.

Template material is common content or formatting that appears on multiple pages of a site. Almost all pages on the web today contain template material to a greater or lesser extent. Common examples include navigation sidebars containing links along the left or right side of the page corporate logos that appear in a uniform location on all pages standard background colors or styles or headers or dropdown menus along the top with links to products, locations, and contact information, banner advertisements and footers containing links to homepages or copyright information. The template mechanism is used to support many purposes particularly navigation, presentation, and branding. There is no single dominant mechanism by which templates appear in web pages. At one extreme is web site design software that allows a user to single-handedly manage a medium-size web site formally editing and applying templates to groups of pages as necessary. At the other extreme is the personal web site in which the owner copies the same fragment of HTML from one page to the next in order to provide a uniform look and feel, and diligently avoids the overhead of changing templates too frequently. Other familiar mechanisms include application servers that implement page templates in code dynamically generated pages that wrap content into template portal servers that arrange content into cells with arbitrary content around them; and content management systems that manage template

1.6 TEMPLATE EXTRACTION

Template extraction is an important technique since templates could heavily impact the performance of other modules such as page classification modules or index builders. Most previous approaches utilize content repetition as a hint for template detection. A block which occurs in many pages is considered to be a template block. For the use of content repetition, they all require a lot of web pages as input. So in practice, these methods run in a batch manner. They start working when enough web pages of a site have been gathered. They detect all template blocks and then wait until another dataset is available. However, caching the web pages often consume lot of storage. Since a newly crawled page cannot be processed until enough pages are collected the delay of data refreshing is huge. So the batch manner is not applicable to schemes in which the speed of data refreshing is concerned. These schemes include news search engines, blog search engines, etc.

1.7 RESTRICTED TOP DOWN MAPPING

Intuitively, in the restricted top-down mapping, besides the insertion and removal operations, the replacement operation of different vertices is also restricted to the leaves of the trees. More formally, we have the following definition.

A top-down mapping M between a tree T_1 and a tree T_2 is said to be restricted top-down only if for every pair $(i_1; i_2) \in M$, such that $t_1[i_1] \neq t_2[i_2]$, there is no descendent of i_1 or i_2 in M, where i_1 and i_2 are non-root nodes of T_1 and T_2 respectively.

1.8 LITERATURE SURVEY

1.8.1 THE VOLUME AND EVOLUTION OF WEB PAGE TEMPLATES

In (David Gibson2005) authors discussed about the evolution of the web pages. Web pages contain a combination of unique content and template material, which is present across multiple pages and used primarily for formatting, navigation, and branding. They study the nature, evolution, and prevalence of these templates on the web. As part of this work, they have developed new randomized algorithms for template extraction that perform approximately twenty times faster than existing approaches with similar quality. Their contributions are primarily focused on measuring the nature and extent of templates on the web.

METHODOLOGY USED

However, in order to perform this task, they also develop efficient algorithms for template extraction which are appropriate for integration into the workflow of a traditional large-scale web crawler. These algorithms are simple to implement, and allow detection and removal of templates on the fly using very little memory footprint per site like DOM based algorithm. Report is on two studies. First randomly sampling two hundred sites from a large crawl containing approximately fifty million sites and two billion pages. Then hand-classifying the site-level sample into seven categories such as personal sites, catalog sites, community sites and so on. Analyze the nature and prevalence of templates within sites belonging to each category. For each site, it creates a uniform random sample of the crawled content from the site of approximately two hundred pages, and studies the commonality of templating across the sample. In this a

novel visualization technique is presented which effectively captures the template structure of an entire site.

TEMPLATES ON TODAY'S WEB

Using crawls from the Internet Archive, studying multiple snapshots of pages from two collections: the hand-classified sites from first study and the sites studied by Ntoulas et al. Initially gather approximately 72K page instances during this study, over 1380 snapshots of time. It shows how template usage has changed over the last 8 years, and offers some thoughts about what these studies portend for the future. The primary conclusions are the following Template volume. According to the studies the volume of template material is 40–50% of the total bytes on the web, and the growth shows no signs of slowing. This has several implications. First as bandwidth, tooling, and browser capabilities increase, there will be ever more complex and costly templates attached to pages. As browsing patterns show, users tend to visit multiple pages when they visit a site, suggesting that a more sophisticated approach to client-side template caching would be in order

Second, for organizations like search engines and archives that store significant amounts of web content, results suggest that documents organized by site and compressed schemes that allow implicit or explicit references to site-level templates will show significant savings. Finally, analytical operations in a domain with inline template extraction will run twice as fast as they do today, rather than spending half their time re-processing the same template bytes.

TEMPLATE TYPES

Different types of web sites show very different templates behavior. While media sites are often presented as examples of aggressive templates, and there are high-profile examples of such sites, they show that the average media site typically small local media outlets in fact use templates far less frequently than the rest of the web. On the other hand, catalog sites dedicated to presenting items favorably to consumers offer templates covering 60% or more of their content.

RATE OF CHANGES

It shows that the duration of the average template is quite similar to the duration of the de-template region of the page. First consider changed pages, and study the distribution of the magnitude of change. They show that this distribution is weighted more heavily towards the tail changes of large magnitude once the template content has been removed, suggesting that pages do in fact change more or less completely with significant frequency.

LINKS, TEXT, AND HTML

The fraction of HTML content, and hyperlinks, that appear in templates are comparable, ranging from about 35% to about 50% over a number of data sets**.** The fraction of debagged content in templates show a somewhat broader range as low as 24% in some cases to as high as 53% in others. The rates of change of these three quantities range from 6 to 8% per year.

There are two key implications. First, the graph structure of the web is increasingly dominated by boilerplate, suggesting that link analysis algorithms require an understanding of templates.

Second, all categories of template usage are on the rise, suggesting that navigation, layout, and publishing of textual content via templates are all important and growing tools in the toolbox of web designing

STRENGTH

1 Templates represent 40-50% of total bytes and content is growing at the rate of 6% every year. The fraction of visible words and fraction of hyperlinks appearing in template is extremely high.

2. The graph structure of the web is increasingly dominated by boiler plate, suggesting the link analysis algorithm requires understanding of templates.

WEAKNESS

1. Templates degrade the performance of web pages and also data mining techniques like clustering and classification.

2. Less accuracy and precision of the existing method

1.8.2 XTRACT: A SYSTEM FOR EXTRACTING DOCUMENT TYPE DESCRIPTOR FROM XML DOCUMENTS

In (Rajiv Rastogi 2000) authors proposed XTRACT, a novel system for inferring a DTD schema for a database of XML documents. Since the DTD syntax incorporates the full expressive power of regular expressions, naive approaches typically fail to produce concise and intuitive DTDs

XML is rapidly emerging as the new standard for data representation and exchange on the Web. An XML document can be accompanied by a Document Type plays the role of a schema for an XML data collection. DTDs contain valuable information on the structure of documents and thus have a crucial role in the efficient storage of XML data, as well as the effective formulation and optimization of XML queries.

DOCUMENT TYPE DESCRIPTOR

A characteristic, however, that distinguishes XML from semi structured data models is the notion of a Document Type Descriptor that may optionally accompany an XML document. A document's DTD serves the role of a schema specifying the internal structure of the document. Essentially, a DTD specifies for every element, the regular expression pattern that sub element sequences of the element need to conform to. DTDs are critical to realizing the promise of XML as the data representation format that enables free interchange of electronic data and integration of related news, products, and services information from disparate data sources.

DTDs play a crucial role in the efficient storage of XML data as well as the formulation, optimization, and processing of queries over a collection of XML documents. For instance, in DTD, information is exploited to generate effective relational schemas, which are subsequently employed to efficiently store and query entire XML documents in a relational database. Frequently occurring the portions of XML documents are stored in a relational system.

XTRACT SYSTEM

In the inference algorithms employed in the XTRACT system, proposes the following novel combination of sophisticated techniques to generate DTD schemas that effectively capture the structure of the input sequences

GENERALIZATION

As a first step, the XTRACT system employs novel heuristic algorithms for finding patterns in each input sequence and replacing them with appropriate regular expressions to produce more general candidate DTDs. The main goal of the generalization step is to judiciously introduce meta characters to produce regular sub expressions that generalize the patterns observed in the input sequences

FACTORING

As a second step, the XTRACT system factors common sub expressions from the generalized candidate DTDs obtained from the generalization step, in order to make them more concise.

MINIMUM DESCRIPTION LENGTH PRINCIPLE

In the final and most important step, the XTRACT system employs Minimum Description Length principle to derive an elegant mechanism for composing a near-optimal DTD schema from the set of candidate DTDs generated by the earlier two steps

Abstractly, MDL ranks each candidate DTD depending on the number of bits required to describe the input collection of sequences in terms of the DTDs .DTDs requiring fewer bits are ranked higher. As a consequence, the optimal DTD according to the MDL principle is the one that is general enough to cover a large subset of the input sequences but, at the same time, captures the structure of the input sequences with a fair amount of detail, so that they can be described easily with few additional bits using the DTD. Thus, the MDL principle provides a formal notion of best DTD that exactly matches our intuition.

STRENGTH

1. Extracts concise and semantically meaningful DTD from xml documents.
2. A number of the DTDs which were correctly identified by XTRACT were fairly complex and contained factors, meta characters and nested regular expression terms.

WEAKNESS

1. Extracted DTD is voluminous and unintitutive.
2. Many XML Document based on relational databases not accompanied with DTD.

1.8.3 EXTRACTING STRUCTURED DATA FROM WEB PAGES

In (Aravind Arasu 2003) authors discussed the method of data extraction from web pages. Extracting structured data from the web pages is clearly very useful, since it enables us to pose complex queries over the data. Extracting structured data has also been recognized as an important sub-problem in information integration systems, which integrate the data present in different web-sites. Therefore, there has been a lot of recent research in the database and Artificial Intelligence communities on the problem of extracting data from web pages sometimes called information extraction problem. This method studies the problem of automatically extracting structured data encoded in a given collection of pages, without any human input like manually generated rules or training sets. For instance, from a collection of pages it would like to extract book tuples, where each tuple consists of the title, the set of authors, the optional list-price, and other attributes.

EXTRACTION OF DATA

An important characteristic of pages belonging to the same site and encoding data of the same schema is that the data encoding is done in a consistent manner across all the pages. For example, the title of the book appears in the beginning followed by the word followed by the author. In other words, the pages are generated using a common template by plugging-in values for the title, list of authors and so on. Most of the information extraction techniques proposed so far and the technique that proposed in this, exploits the template based encoding for extracting data from the pages. Specifically, the techniques use either a partial or complete knowledge of the template used to generate the pages, to extract the data. For example, the price of a book can be extracted by retrieving the text immediately following the template-text book price

The main goal is to deduce the template without any human input, and use the deduced template to extract data. There are at least two reasons why absence of human input is beneficial. First, human input is time consuming and error-prone. A single web page used in training could potentially contain a large number of data values of interest, and the human trainer has to identify each one of them; and the training could require many such human annotated pages. Second, many collection of pages are semi-structured, and contain optional attributes. If an optional attribute appears very rarely, it is possible for the human trainer to miss the optional attribute altogether. Both the above problems are further aggravated by the fact that, in practice, templates changes very frequently, requiring repeated human intervention.

MODEL FOR PAGE CREATION

There are two fundamental challenges in automatically deducing the template. First and foremost there is no obvious way of differentiating between text that is part of template and text that is part of data. Any word could be part of template, or data or both. Note that it is not necessary for a word that is part of template to occur in every page. Conversely, a common English word likes 'is 'could occur as part of data in every input page. Second, the schema of data in pages are usually not a flat set of attributes, but is more complex and semi-structured. The schema could contain non-atomic attributes that are sets of values. The existence of complex schema makes both the tasks of definition and automatic recognition of template harder. In fact the existence of complex schema makes problem very closely related to the problem of regular expression inference which is known to be very hard.

Page	A	B	C
1	MySystem	Aron	Null
2	Godel	Douglas	20.00

Figure 1.1 Extracted Data

```
<html>

<body>

<h1>Book</h1> C Programming
Language by Brian Kernighan

</body>

</html>
```

Figure 1.2 Templates and Page Structure

Figure1.2 shows template and page structure of extracted data. They make two clarifications regarding to assumptions and goals. First, their goal is not to try to semantically name the extracted data. Assuming that renaming, for example, attribute A as TITLE is done as a post-processing step, possibly with human help. Second, they assume that the input pages conform to a common schema and template. They do not consider the problem of automatically obtaining such pages from web sites. It is reasonably easy for a human to identify web collections of interest that have a common schema and then run a crawler to gather the pages.

STRENGTH

1. Good in extracting data from web pages.

2. Does not fail to extract if assumptions are wrong.

WEAKNESS

1. Not fit for extracting data from heterogeneous web pages.
2. Less feasible in generating large databases

1.8.4 AUTOMATIC WEB NEWS EXTRACTION USING TREE EDIT DISTANCE

In (Davide Castro Resis 2004) authors proposed a domain based approach for web data extraction . Devising generic methods for extracting Web data is a complex task, since the Web is very heterogeneous and there are no rigid guidelines on how to build HTML pages and how to declare the implicit structure of the Web pages. Thus, in order to develop effective methods for extracting Web data in a precise and completely automatic manner, it is usually required to take into account specific characteristics of the domain of interest. One of such domains is that of on-line newspapers and news portals on the Web, which have become one of the most important sources of up-to-date information. Indeed, there are thousands of sites that provide daily news in very distinct formats and there is a growing need for tools that will allow individuals to access and keep track of this information in a automatic manner.

DOMAIN BASED APPROACH

A domain-oriented approach is presented for Web data extraction and discussing its application to automatically extracting news from Web sites. This approach is based on the concept of tree-edit distance and allows not only the extraction of relevant text passages from the pages of a given Web site, but also the fetching of the entire Web site content, the identification of the pages of interest the pages that actually present the

news and the extraction of the relevant text passages discarding non-useful material such as banners, menus, and links. To support this approach, a highly efficient tree structure analysis algorithm that outperforms is used for practical purposes with the best results on tree-edit distance calculation in the literature. The approach is tested 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

The approach is developed for finding and extracting data of interest from Web pages is based on the analysis of the structure of these target pages. More precisely, by evaluating the structural similarities between pages in a target site it is easy to perform tasks such as grouping together pages with similar structure to form page clusters and finding a generic representation of the structure of the pages within a cluster.

TREE EDIT DISTANCE

Since the structure of a Web page can be nicely described by a tree e.g. a DOM tree, so this is based on the concept of tree edit distance to evaluate the structural similarities between pages. Intuitively, the edit distance between two trees TA and TB is the cost associated with the minimal set of operations needed to transform TA into TB. Trees are one of the most common data structures used in computer science. Formally, they are defined as directed acyclic simple graphs.

Web news extraction approach that relies on the RTDM algorithm to identify relevant text passages containing news and their components, extract them and discard useless material such as banners, links, etc. The approach has basically two main tasks: (1) the crawling of news portals to fetch the pages of interest and (2) the extraction of the news from the HTML pages collected.

In the case of news, templates are filled by journalists, usually through the use of specific Web applications or some database interface. Each field of a template (e.g., a news title) considered as data-rich object. Ideally, the extractors generated by the approach should be able to identify each one of these data-rich objects, and discover, among them, which ones correspond to the title and the body of the news.

According to this, the extraction task is performed in four distinct steps: (1) page clustering, (2) extraction pattern generation,(3) data matching and (4) data labelling.

STRENGTH

1. Solves problem like structure based classification, data labeling etc

WEAKNESS

1. Not suitable to other domain applications whose page schema of data is complex

1.8.5 AUTOMATIC EXTRACTION OF DYNAMIC RECORD SECTIONS FROM SEARCH ENGINE RESULT PAGE

In (Clement Yu 2009) authors proposed how to extract the static and dynamic contents of web pages. A typical search engine result page contains static, semi dynamic and dynamic contents. In this paper, static contents refer to the portion that is query independent and they are identical on the result page of every query. Dynamic

contents are the Section Records (SR) retrieved in response to a query. Each record is a semantically complete data unit corresponding to a retrieved entity. A record typically consists of a link to a retrieved Web page or database record and some pertinent information. If there is no ambiguity, the word record also refers to Section Record Result. Semi-dynamic contents are those that may be affected by different queries but are generally independent of the content of any specific query.

Intuitively, a dynamic section on a search result page is a set of all Section Record (SR) that appear consecutively and have certain common features such as a common header and a common display format. Many search engines produce result pages with multiple dynamic sections. For example, some search engines categorize or cluster search results and some search engines display regular search results and sponsored links in different dynamic sections. A significant percentage of the search engines return result pages with multiple dynamic sections. For example, 19 out of the 100 search engines from the dataset produce result pages with multiple dynamic sections

RECORD DIVERSITY

In general, complete data extraction from web pages including result pages returned from search engines may consist of three tasks. The first is section extraction extracts all the sections from each page; the second is record extraction, extract the records within each section; and the third is data annotation ,identifies and annotates each data unit within each record. Existing work on data extraction wrapper generation has been mostly focused on record extraction and some work on data annotation has also been reported.

It is investigated how to automatically extract all dynamic sections as well as SRRs within each dynamic section from search result pages. The emphasis is on dynamic section extraction. Static and semi-dynamic contents are utilized to help identify the boundaries of different dynamic sections. For the rest of this when there is no confusion, dynamic section and section will be used interchangeably. An important requirement for this method is to maintain the section-record relationship the extracted SRRs should be grouped by section. This requires that the sections be explicitly extracted. The benefit of keeping the section-record relationship is to make it easier to use them later as different applications may be interested in the SRRs in different sections.

MULTIPLE SECTION EXTRACTION

The wrapper generation algorithm will be called **M**ultiple **S**ection **E**xtraction (MSE). The input to MSE is a set of n sample result pages from a search engine (SE). These result pages are returned from SE in response to *n* different queries. The output of MSE is a wrapper or a set of rules for extracting all dynamic sections as well as all SRRs within them. Figure 1.3 shows working of system overview described below

Multiple record sections consist of the following steps:

1. Render each result page and extract its content lines by a pre-order traversal of the DOM trees.
2. Extract record sections that contain multiple records from each result page. These sections called multi-record sections and denoted as multiple records and the algorithm for extracting records will be denoted as multiple record sections.

3. Identify dynamic sections by applying algorithm dynamic section extraction. In order to perform this task, we need to identify candidate section boundary mark on each page.
4. Refine multiple records and dynamic section by analyzing their relationships. By comparing this, records containing static contents can be identified and discarded, and some incorrect boundaries of records can be eliminated.
5. Many sections contain more than three records. As a result, even a single record can be extracted from a section.
6. Check identified sections and records to see if they are correctly identified.
7. Group extracted section instances from all sample result pages into clusters such that each cluster corresponds to the same section schema of the result page schema of the search engine.
8. Generate the extraction wrapper for each section schema based on the section instances in the corresponding cluster.
9. Generate section families, each of which is a class of section schemas that share some common features.

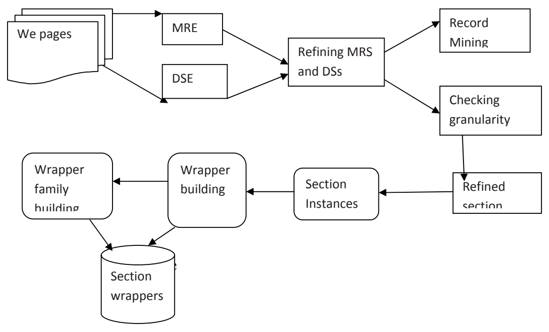


Figure 1.3 System Overview

STRENGTH

1. Method is automatically useful for all web applications that need to interact with search engines.

2. Accuracy is high in extracting the records.

WEAKNESS

1. Difficult to identify the section and boundary and need additional experiment for this method.

2. Requirement is that section must contain at least three records

CHAPTER 2

IMPLEMENTATION

2.1 EXISTING SYSTEM

World Wide Web (WWW) is widely used to publish and access information on the Internet. 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 common templates with contents. For machines, the unknown templates are considered harmful because they degrade the accuracy and performance due to the irrelevant terms in templates. Thus, template detection and extraction techniques have received a lot of attention recently to improve the performance of web applications, such as data integration, search engines, classification of web documents the assumption of all documents being generated from a single common template, solutions for this problem are applicable only when all documents are guaranteed to conform to a common template

However, in real applications, it is not trivial to classify massively crawled documents into homogeneous partitions in order to use these techniques. Since subtle changes in scripts or CGI parameters may result in a significant difference cannot simply group the web documents by URL and apply these methods for each group separately. Since an HTML document can be naturally represented with a Document Object Model (DOM) tree, web documents are considered as trees and many existing

similarity measures for trees have been investigated for clustering is very expensive with tree-related distance measures.

Advantages

1. Improves the performance of web applications like data integration, search engine, clustering.

2. Improves efficiency and scalability of clustering process

Disadvantage

1. Execution time is more.

2.2 PROPOSED SYSTEM

The main issue of the proposed approach is to extract the template from the duplicate web pages in order to obtain a better view of the information in a single context of template. There have been several approaches presented for this purpose of the web template extraction. The HTML documents is converted into DOM model and each document model is represented into their respective matrix representation for the extraction process.

Initially the RTDM algorithm is used for extracting the template from the web pages. Next a naive agglomerative algorithm named TEXT-MDL approach is used. In order to reduce the time complexity it is needed to employ the method of minhash approach to the MDL principle. Then proposed the TEXT-MAXHASH algorithm which a clustering algorithm with both MinHash signatures and Heuristic to reduce the search space. It requires the length of the signature as an input parameter. Finally introduced the most efficient approach for this problem of template extraction namely the Locality

Sensitive Hashing approach. Evaluating the performance of the proposed approach with the existing methods based on a set of performance metrics such as time, entropy, approximation ratio, precision and recall. The experimental result shows that proposed approach works better than the existing with high performance on the speed and accuracy, etc.

Advantages of proposed system

1. Performs probabilistic dimension reduction of high dimensional data

2.3 PROBLEM DEFINITION

Template Extraction from the web pages confirming to common site is quite achievable. However in real application it is trivial to extract the templates from the collections of heterogeneous web documents and classifying them. Thus correctness of extracted templates depends on the quality of clustering.

2.4 PROJECT OVERVIEW

The enhanced approach is the method of clustering based on the similarity. A similarity based clustering approach Locality Sensitive Hashing based method is employed. It starts with a random projection operation that maps a data point from a high-dimensional point to a low-dimensional subspace. First, to note which points are close to our query points. Second, is to create projections from a number of different directions and keep track of the nearby points. Keep a list of these found points and note the points that appear close to each other in more than one projection. Part of the art of solving this unlike conventional computer hashes that are designed to return exact matches in O (1) time, an LSH algorithm uses dot products with random vectors to quickly find nearest neighbors. LSH provides a probabilistic guarantee that it will return the correct answer. In systems that have other sources of error (perhaps due to mislabeled data) one can reduce the LSH error below the error due to other sources, while significantly improving

The Locality Sensitive Hashing algorithm has been applied successfully to quickly find nearest neighbors in very large database. Instead of finding exact matches as conventional hashes would .Locality Sensitive Hashing takes into account the locality of the points so that nearby points remain nearby.

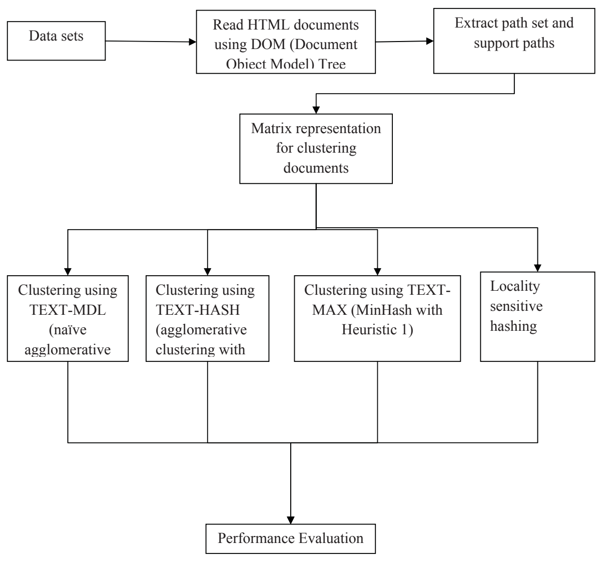

Data sets
Read HTML documents using DOM (Document Object Model) Tree
Extract path set and support paths
Matrix representation for clustering documents
Clustering using TEXT-MDL (naïve agglomerative
Clustering using TEXT-HASH (agglomerative clustering with
Clustering using TEXT-MAX (MinHash with Heuristic 1)
Locality sensitive hashing
Performance Evaluation

Figure 2.1 Template Extraction

2.5 MODULES

2.5.1 DOM TREE CONSTRUCTION AND MATRIX REPRESENTATION

Document Object Model (DOM) is the tree structure of the respective html documents. Initially have to define the path for the set of documents. Since the document node is a virtual node shared by every document, so 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 d_i, minimum support threshold t_{di} is assumed. The thresholds t_{di} and t_{dj} of two distinct documents d_i and d_j, respectively, may be different

To represent a clustering information C = {c_1, c_2, . . . $c_{m\}}$ for D, use a pair of matrices M_T and M_D, where M_T represents the information of each cluster with its template paths and M_D denotes the information of each cluster with its member documents. If the value at a cell (i, j) in M_T is 1, it means that a path pi is a template path of a cluster cj (i.e., p_i □ T_j)

2.5.2 CLUSTERING USING TEXT MDL

TEXT-MDL is an agglomerative hierarchical clustering algorithm which starts with each input document as an individual cluster. When a pair of clusters are 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.

2.5.3 CLUSTERING USING TEXT MINHASH

The TEXT-MINHASH approach is an extended version of the TEXT MDL approach with the MinHash function. This approach is to modify the approach of clustering by reducing the time complexity of the approach. To compute the MDL cost of each clustering quickly, it is needed 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, for given collection of sets X = {S_1, . . . , S_k}, MinHash estimates the probabilities needed to compute the MDL cost. The MDL cost can be calculated using by MinHash, by knowing the sup(p_k, D_i) for each p_k and can decide the optimal T_i.

2.5.4 CLUSTERING USING TEXT HASH

TEXT-MAXHASH is the clustering algorithm with both MinHash signatures and Heuristic to reduce the search space. It requires the length of the signature as an input parameter. When clusters are merged hierarchically, select two clusters which maximize the reduction of the MDL cost by merging them. Given a cluster c_i, if a cluster c_j maximizes the reduction of the MDL cost is c_j the nearest cluster of c_i. Then, given three clusters ci, cj, and ck, if Jaccard's coefficient between ci and cj is greater than that between ci and ck, it is assumed that the reduction of the MDL cost by merging ci and cj will be greater than that of ci and ck.

2.5.5 LOCALITY SENSITIVE HASHING

Locality Sensitive Hashing Function helps to find the similarity between the web pages easily. . The first large-scale Web search engine, used random selections (similarly to LSH) to test the similarity of pages. If the shingles of the new page match shingles from the database, then it is likely that the new page bears a strong resemblance to an existing page. The nearest-neighbor solution is important because Web pages are surrounded by navigational and other information that changes from site to site. An approximate solution to this problem is desired, especially when balanced with the computational savings of a solution like LSH.

First, is to note which points are close to our query points. Second, creating projections from a number of different directions and keeping track of the nearby points. Then keep a list of those points which closer to each other in more than one projection.

RANNDOM PROJECTION: DOT PRODUCTS

At the core of LSH is the scalar projection (or dot product), given by $h(\vec{v}) = \vec{v} \cdot \vec{x}$, where $\vec{v}$ is a query point in a high-dimensional space, and $\vec{x}$ is a vector with components that are selected at random from a Gaussian distribution, for example N(0, 1). The scalar projection is then quantized into a set of hash bins, with the intention that nearby items in the original space will fall into the same bin. The resulting full hash function is given by

$$h^{x,b}(\vec{v}) = \left\lfloor \frac{\vec{x} \cdot \vec{v} + b}{w} \right\rfloor$$

Where ⌊.⌋ is the floor operation, w is the width of each quantization bin, and b is a random variable uniformly distributed between 0 and w that makes the quantization error easier to analyze, with no loss in performance. For the projection operator to serve our purposes, it must project nearby points to positions that are close together.

RANDOM PROJECTIONS: K DOT PRODUCTS

It further magnifies the difference between $P1$ and $P2$, by performing k dot products in parallel. This increases the ratio of the probabilities (given above) that points at different separations will fall into the same quantization bin, since $(P1/P2)k > (P1/P2)$. The resulting projection is obtained by performing the k independent dot products to transform the query point $\vec{v}$ into k real numbers. As with the scalar (dot product) projection, quantize the k inner products per (1) with the intention that similar points will fall in the same bucket in all dimensions. Increasing the quantization bucket width w will increase the number of points that fall into each bucket. To obtain the final nearest neighbor result it is needed to perform a linear search through all the points that fall into the same bucket as the query, so varying w effects a trade-off between a larger table with a smaller final linear search, or a more compact table with more points to consider in the final search.

HASH IMPLEMENTATION

The process of projection and quantization places each data point in a hash bucket described by k integer indices. Since this k-dimensional space is sparse, use a conventional (exact) hashing method to efficiently find when points fall into common buckets. A naïve search to find reference points in the same bucket as the query point could easily take O(log N) operations, but can be reduced to O(1) by using a pair of conventional hash functions. First, use a conventional hash to map the k-dimensional quantized projection into a single linear index by computing the following

$$T_1 = \left(\sum_i H_i k_i \right) \mod P_1$$

Where Hi are integer weights and $P1$ is the hash table size (a large prime number). The goal of the hash table is to put each distinct point in the k-dimensional space into a separate one of the $P1$ table entries to avoid collision in which unrelated points hash to the same value. Although a well constructed hash will distribute entries quite uniformly, as the table gets smaller the risk of unrelated points colliding naturally increases.

To accommodate above problem, use a second hash $T2$ of the k-dimensional points to check that points retrieved from the hash table do indeed match our query. $T2$ has the same form as $T1$ but uses different weights and size

CHAPTER 3

RESULTS

3.1 SYSTEM SPECIFICATION

3.1.1 HARDWARE REQUIREMENTS

Processor	: Pentium III and above
Clock speed	: 550 MHz
Hard Disk	: 80GB
RAM	: 1GB
Cache	: 512KB
Monitor	: Color Monitor
Keyboard	: 104 Keys
Mouse	: 3 Buttons

3.1.2 SOFTWARE REQUIREMENTS

Operating system	: Windows XP/Vista
Language	: Java
Database	: Microsoft SQL Server
Database drivers	: ODBC driver, SQL Server

3.1.3 SOFTWARE DESCRIPTION

JAVA

Java is related to C++, which is a direct descendant of C. The trouble with C and C++ is that they are designed to be compiled for a specific target. But Java is a portable, platform-independent language that could be used to produce code that would run on a variety of CPUs under differing environments. Java can be used to create two types of programs: applications and applets. An application is a program that runs on our computer, under the operating system of that computer. An applet is an application designed to be transmitted over the Internet and executed by a Java-compatible Web browser. Java is Simple, Secure, Portable, Object-oriented, Robust, Multithreaded, Architectural-neutral, Interpreted, High Performance, Distributed, and Dynamic.

SQL SERVER 2000

INTRODUCTION TO MICROSOFT SQL SERVER 2000

Microsoft SQL Server 2000 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. The following are more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager.

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system traces that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problem.

3.2 EXPERIMENTS AND RESULTS

The proposed system is evaluated with the existing methods. Locality Sensitive Hashing uses dot products with random vectors to quickly find the nearest neighbors. The Locality Sensitive Hashing algorithm is implemented in java, where Graphical User Interface is user friendly and can discover the dense units. The back end Microsoft SQL server is used to store and retrieve the data. The Locality Sensitive Hashing uses HTML document which contain multiple tags.

To evaluate the quality of clusters Precision and Recall are used. Precision is defined as the percentage of the data points in this clusters that really belong to the known cluster. Recall is defined as the percentage of the data points in a known cluster that are identified in this cluster

.

The performance evaluation of proposed algorithm with the existing methods is shown below. The below screenshot shows that the LSH approach performs with a greater speed by consuming very less time for the template extraction than the previous approaches. The figure 6.1 shows the performance evaluation of proposed and existing methods. The figure 6.2 shows the utilization of main memory and disk space. When Minimum description is used it occupies 6000 bytes in main memory and when Locality Sensitive Hashing is used it only occupies 1000 bytes.

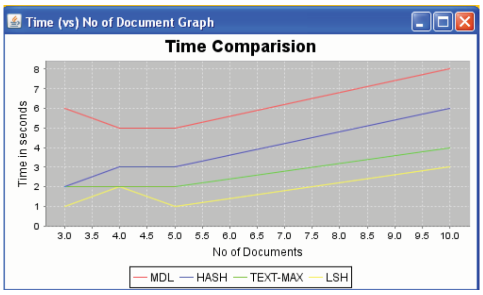


Figure 3.1 Comparision of Performance

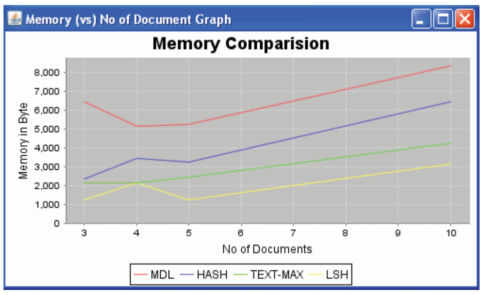


Figure 3.2 Comparision of Memory

3.3 CONCLUSION AND FUTURE WORKS

An enhanced approach is based on the method of clustering based on the similarity. The similarity based clustering approach is called Locality Sensitive Hashing based. It starts with a random projection operation that maps a data point from a high-dimensional point to a low-dimensional subspace. First, note the points which are close to query points. Second, is to create projections from a number of different directions and keep track of the nearby points. It keeps a list of the points and notes the points that appear close to each other in more than one projection. Part of the art solving this unlike conventional computer hashes that are designed to return exact matches in O (1) time, a locality sensitive hashing algorithm uses dot products with random vectors to quickly find nearest neighbors. Locality sensitive hashing provides a probabilistic guarantee that it will return the correct answer. In systems that have other sources of error one can reduce the LSH error below the error due to other sources, while significantly improving.

The future work is to tackle the problem of mining the dynamic large database which contains the unstructured data like xml document contents which require format conversion. The method work can be extended to extract the template from unstructured xml documents and converting to DTD type and extracting the relevant information.

APPENDIX 1

SOURCE CODE

MATRIX

```
public class Matrix
{
int cunt=0,ini=0;
int dd[]={0,0,1};
int dd[]=null;
int arr[][]=null;
String id="",item="";
database_conn db=null;
boolean bt=true;
String doc[]=null;
public Matrix()
try
{
db=new database_conn();
db.st.executeUpdate("if object_id('ME') is not null drop table ME");
File ff=new File("File");
String file[]=ff.list();
```

```
System.out.println("length "+file.length);
dd=new int[file.length];
for(int q=0;q<dd.length;q++)
{
 int rad=(int) (Math.random()*(2-0))+0;
 dd[q]=rad;
  }
  doc=new String[file.length];
  for(int r=1;r<=file.length;r++)
  doc[(r-1)]="d"+r;
  String qur="create table ME( id int,";
  for(int w=0;w<doc.length;w++)
   {
   if(w<((doc.length)-1))
   {
   qur+=doc[w]+" "+"int"+",";
   }
   else
   {
   qur+=doc[w]+" "+"int"+")";
   }
   }
```

```
System.out.println("qur "+qur);
db.st2.executeUpdate(qur);
qur="";
ResultSet rd1=db.st.executeQuery("select count(*) from Count");
if(rd1.next())
ini=rd1.getInt(1);
arr=new int[file.length][ini];
int y=0;
String lin="",linn="";
for(int i=0;i<dd.length;i++)
{
y=0;
System.out.println("file "+"["+i+"]"+file[i]);
FileReader fd=new FileReader("File\\"+file[i]);
BufferedReader br=new BufferedReader(fd);
while((lin=br.readLine())!=null)
{
linn+=lin+"\n";
}
String ld[]=linn.split("\n");
linn="";
ResultSet rd=db.st.executeQuery("select * from Count order id");
```

```
 while(rd.next())
 {
  id=""+rd.getString(1);
 item=(""+rd.getString(2)).trim();
 cunt=rd.getInt(3);
 bt=true;
 System.out.println("item "+item);
 if(cunt>dd[i])
 {
 System.out.println("if item "+item);
  for(int q=0;q<ld.length;q++)
 {
  if((ld[q].trim()).equalsIgnoreCase(item))
 {
  arr[i][y]=1;
  System.out.println("equals "+i+y);
  bt=false;
 }
 else
 {
 if(bt==true)
 {
```

```
  arr[i][y]=0;
 System.out.println("not equals "+i+y);
 bt=false
  else
 {
 arr[i][y]=0;
 }
 System.out.println("arr "+"["+i+"]"+"["+y+"]"+arr[i][y]);
 }
 }
 System.out.println("Matrix ");
 String qur1="insert into ME values(";
 for(int a=0;a<ini;a++)
 {
  qur1+=(a+1)+",";
 for(int a1=0;a1<doc.length;a1++)
 {
 if(a1<((doc.length)-1))
 qur1+=arr[a1][a]+",";
 else
 qur1+=arr[a1][a]+")";
}
```

```
 System.out.println("qur1 "+qur1);
 db.st3.executeUpdate(qur1);
 qur1="insert into ME values(";
 }
 ViewTable vt=new ViewTable();
 }
 catch(Exception ed)
{
 ed.printStackTrace();
}
}
public static void main(String ar[])
{
Matrix mt=new Matrix();
}
```

CLUSTER

```
public class Cluster extend
{
database_conn db=null;
Vector doc=new Vector();
```

```
FileWriter wr=null;
BufferedWriter bw=null;
Vector path=new Vector();
int no,d1,d2,d3,d4;
FileWriter fw1=null;
BufferedWriter bw1=null;
FileWriter fw2=null;
BufferedWriter bw2=null;
int val=1;
String tt="";
static JTextArea ta=null;
JScrollPane sp=null;
JPanel p1=new JPanel(null);
int count[]=null;
JButton b1=new JButton("Exit");
Vector ct=new Vector();
long time=0,time1=0,time2=0;
double mem1=0,mem2=0,memory=0;
 public Cluster()
{
 super("MDL Cluster Process");
  try
 {
mem1=Runtime.getRuntime().totalMemory();
fw1=new FileWriter("Ctime.txt",true);
bw1=new BufferedWriter(fw1);
fw2=new FileWriter("Cmem.txt",true);
bw2=new BufferedWriter(fw2);
time = System.currentTimeMillis()/1000;
wr=new FileWriter("Clus");
bw=new BufferedWriter(wr);
ta=new JTextArea();
ta.setForeground(Color.blue);
ta.setFont(new Font("serif",Font.BOLD,15));
sp=new JScrollPane(ta);
p1.setBackground(Color.orange);
p1.add(sp);
p1.add(b1);
sp.setBounds(100,100,400,400);
b1.setBounds(200,500,100,30);
getContentPane().add(p1);
setSize(600,600);
Thread.sleep(5000);
setVisible(true);
```

```
File fd=new File("File");
String nam[]=fd.list();
int tot=nam.length;
count=new int[tot-1];
db=new database_conn();
String qur1="select count(*) from ME where d1="+val+" and ";
for(int q=1;q<tot;q++)
{
qur1+="d"+(q+1)+"="+val;
System.out.println("qur1 "+qur1);
ResultSet rd=db.st.executeQuery(qur1);
if(rd.next())
count[q-1]=rd.getInt(1);
qur1="select count(*) from ME where d1="+val+" and ";
if(ct.contains(count[q-1])){}
else
{
ct.add(count[q-1]);
}
}
for(int e=0;e<count.length;e++)
{
```

```
System.out.println("count "+"["+e+"]"+count[e]);
}
qur1="select count(*) from ME where d1="+val+" and ";
String mt="",mm="";
int qt=1;
for(int q=1;q<tot;q++)
{
qur1+="d"+(q+1)+"="+val;
System.out.println("qur2 "+qur1);
ResultSet rd=db.st.executeQuery(qur1);
if(rd.next())
{
 int cunt=rd.getInt(1);
 for(int k=0;k<ct.size();k++)
 {
 int cunt1=Integer.parseInt(""+ct.elementAt(k));
  if(cunt==cunt1)
 {
 mt=k+":"+"d"+(qt)+" "+"d"+(q+1);
 mm+=mt+",";
  }
  }
```

```
 }
 qur1="select count(*) from ME where d1="+val+" and ";
  }
```

LSH FUNCTION

```
public LSH()
{
super("LSH");
try
{
 time=System.currentTimeMillis()/1000;
 db=new database_conn();
 fw1=new FileWriter("Ltime.txt",true);
 bw1=new BufferedWriter(fw1);
 b1=new JButton("Exit");
 ta.setForeground(Color.blue);
 ta.setFont(new Font("serif",Font.BOLD,15));
 p.setBackground(Color.orange);
 p.add(sp);
 p.add(b1);
getContentPane().add(p);
sp.setBounds(50,100,500,300);
b1.setBounds(200,500,100,30);
setLocation((dd.width-600)/2,((dd.height-600)/2)-10);
setSize(600,600);
setVisible(true);
 get();
 b1.addActionListener(this);
 }
 catch(Exception ed)
 {
  ed.printStackTrace();
 }
 }
 public void actionPerformed(ActionEvent ae)
 {
  if(ae.getSource()==b1)
 {
  setVisible(false);
  }
  }
  public static void main(String ar[])
```

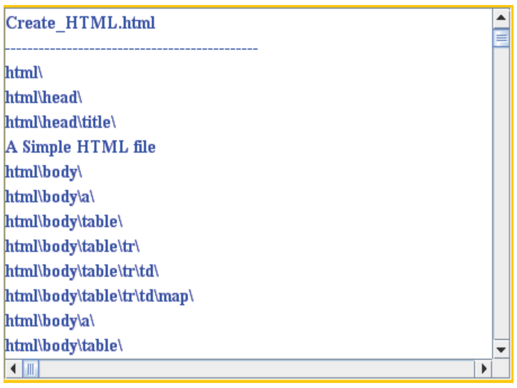
```
{
 LSH ls=new LSH();
}
 public void get()
 {
 try
 {
 ta.append("LSH Process\n");
 ta.append(" Document Clustering \n");
 File in=new File("Inputs");
 String size[]=in.list();
 tot=size.length;
 String size1[]=new String[((size.length)-1)];
 dist=new int[size.length];
 res=new double[((size.length)-1)];
 res1=new String[((size.length)-1)];
 res2=new String[((size.length)-1)];
 sort=new double[((size.length)-1)];
 ran =(int) (Math.random()*(((size.length)-1)-0))+0;
 int w=0;
 System.out.println("ran "+ran);
 for(int k=0;k<size.length;k++)
```

```
 {
  System.out.println("size "+"["+k+"]"+size[k]);
  System.out.println("select sum(d"+((k+1))+") from ME");
  ResultSet rd=db.st.executeQuery("select sum(d"+((k+1))+") from ME");
  if(rd.next())
  {
   dist[k]=rd.getInt(1);
   System.out.println("dist "+"["+k+"]"+dist[k]);
   w+=dist[k];
  if(ran==k)
  {
   query=dist[k];
  }
  }
  }
 Catch(Exception ed)
 {
 }}// end of function he
```

APPENDIX 2

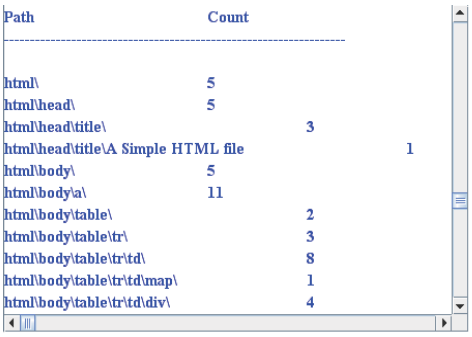
SNAPSHOTS

The following snapshots show DOM tree of the HTML document and corresponding support count of each document



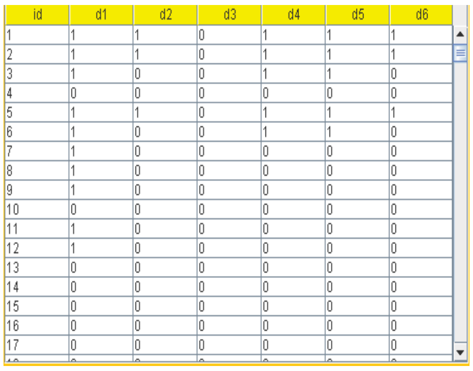
```
Create_HTML.html
----------------------------------------------
html\
html\head\
html\head\title\
A Simple HTML file
html\body\
html\body\a\
html\body\table\
html\body\table\tr\
html\body\table\tr\td\
html\body\table\tr\td\map\
html\body\a\
html\body\table\
```

Figure A.2.1 DOM Tree Representation



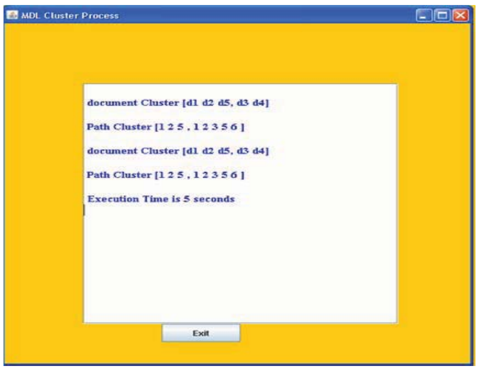
```
Path                          Count
-----------------------------------------------------------------

html\                         5
html\head\                    5
html\head\title\                           3
html\head\title\A Simple HTML file                    1
html\body\                    5
html\body\a\                  11
html\body\table\                           2
html\body\table\tr\                        3
html\body\table\tr\td\                     8
html\body\table\tr\td\map\                 1
html\body\table\tr\td\div\                 4
```

Figure A.2.2 Support Count of HTML Document



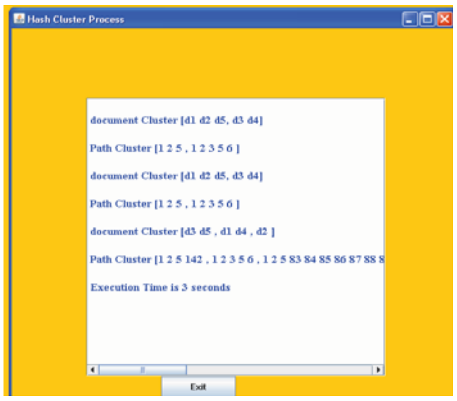
id	d1	d2	d3	d4	d5	d6
1	1	1	0	1	1	1
2	1	1	0	1	1	1
3	1	0	0	1	1	0
4	0	0	0	0	0	0
5	1	1	0	1	1	1
6	1	0	0	1	1	0
7	1	0	0	0	0	0
8	1	0	0	0	0	0
9	1	0	0	0	0	0
10	0	0	0	0	0	0
11	1	0	0	0	0	0
12	1	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0

Figure A.2.3 Matrix Representation of HTML Documents



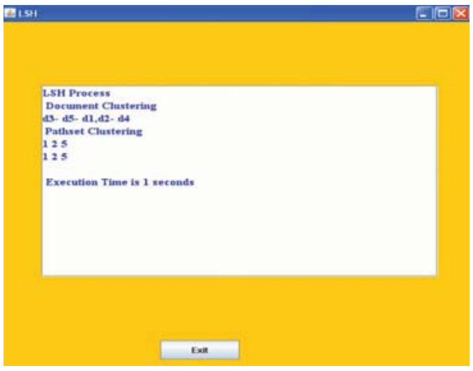
```
document Cluster [d1 d2 d5, d3 d4]
Path Cluster [1 2 5 , 1 2 3 5 6 ]
document Cluster [d1 d2 d5, d3 d4]
Path Cluster [1 2 5 , 1 2 3 5 6 ]
Execution Time is 5 seconds
```

Figure A.2.4 MDL Clustering Process



```
document Cluster [d1 d2 d5, d3 d4]
Path Cluster [1 2 5 , 1 2 3 5 6 ]
document Cluster [d1 d2 d5, d3 d4]
Path Cluster [1 2 5 , 1 2 3 5 6 ]
document Cluster [d3 d5 , d1 d4 , d2 ]
Path Cluster [1 2 5 142 , 1 2 3 5 6 , 1 2 5 83 84 85 86 87 88 8
Execution Time is 3 seconds
```

Figure A.2.5 Hash Clustering Process



```
LSH Process
 Document Clustering
d3- d5- d1,d2- d4
 Pathset Clustering
1 2 5
1 2 5

 Execution Time is 1 seconds
```

Figure A.2.6 Locality Sensitive Hashing Process

REFERENCES

1. Arasu, A., and Garcia-Molina,H. "Extracting Structured Data fromWeb Pages,"

Proc. ACM SIGMOD, 2003.

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

3. AZ 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.

4. DChakrabarti, R Kumar, and K Punera,"Page-Level Template Detection via Isotonic Smoothing," Proc. 16th Int'l Conf. WorldWide Web (WWW), 2007.

5. ZChen, F Korn, N Koudas, and S Muithukrishnan, "Selectivity Estimation for Boolean Queries," Proc. ACM SIGMOD-SIGACT SIGART Symp. Principles of Database Systems (PODS), 2000.

6. J Cho and U Schonfeld,"RankmassCrawler:A Crawler with High Personalized Pagerank Coverage Guarantee," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2007.

7. V Crescenzi, P Merialdo, and P Missier, "Clustering Web Pages Based on Their Structure," Data and Knowledge Eng., vol. 54, pp. 279-299, 2005.

8. V Crescenzi, G Mecca, and P Merialdo, "Roadrunner: Towards Automatic Data Extraction from Large Web Sites," Proc. 27th Int'lConf. Very Large Data Bases (VLDB), 2001.

9. M deCastro Reis,PBGolgher, AS da Silva, and A HFLaender, "AutomaticWeb News Extraction Using Tree Edit Distance,"Proc. 13th Int'l Conf. World Wide Web (WWW), 2004.

10. IS Dhillon, S Mallela, and DS Modha, "Information-Theoretic Co-Clustering," Proc. ACM SIGKDD, 2003.

11. M N Garofalakis, AGionis,RRastogi, SSeshadri, and KShim, "Xtract:ASystemfor Extracting Document Type Descriptors from Xml Documents" PrACM SIGMOD, 2000.

12. DGibson, KPunera, and ATomkins, "The Volume and Evolution of Web Page Templates," Proc. 14th Int'l Conf. WorldWide Web (WWW), 2005.

13. K Lerman, L Getoor, S Minton, and C Knoblock, "Using the Structure of Web Sites for Automatic Segmentation of Tables," Proc. ACM SIGMOD, 2004.

14. B Long,Z Zhang, and PS Yu, "Co-Clustering by BlockValueDecomposition," Proc. ACM SIGKDD, 2005.

15. F Pan, X Zhang, and W Wang, "Crd: Fast Co-Clustering on Large Data Sets Utilizing Sampling-Based Matrix Decomposition," Proc. ACM SIGMOD, 2008.

16. KVieira, AS da Silva, N Pinto, ES de Moura, JM B Cavalcanti, and J Freire, "A Fast and Robust Method for Web Page Template Detection and Removal," Proc. ACM Int'l Conf. Information and Knowledge Management (CIKM), 2006.

17. Y Zhai and B Liu, "Web Data Extraction Based on Partial Tree Alignment," Proc. 14th Int'l Conf. World Wide Web (WWW), 2005.

18. H Zhao, W Meng, and C Yu, "Automatic Extraction of Dynamic RecordSectionsfrom Search Engine Result Pages," Proc. 32nd Int'lConf. Very Large Data Bases (VLDB), 2006.

19. H Zhao, W Meng, Z Wu, V Raghavan, and C Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc. 14th Int'l Conf. World Wide Web (WWW), 2005

20. S Zheng, D Wu, R Song, and JR. Wen, "Joint Optimization of Wrapper Generation and Template Detection," Proc. ACM SIGKDD, 2007.

LIST OF PUBLICATIONS

1. Gayatri, S., Betty, P. "LOCALITY SENSITIVE HASHING SIMILARITY BASED TEMPLATE EXTRACTION FROM HETEROGENEOUS WEB PAGES**",** Bannari Amman Institute of Technology, 21st February 2013.

2. Gayatri, S., Betty, P. "OPTIMIZED TEMPLATE BASED DETECTION MODEL FOR HETEROGENEOUS WEB PAGES ", Government College of Technology, 11th March 2013.