Content Based Web Search And Information Extraction From Heterogeneous Websites Using Ontology

E. Suganya, S. Vijayarani

Abstract: The main purpose of this research work is to evolve a tool which searches the web and extracts the required information from educational institutions websites. A single university/college website contains many different web pages which contains home page, facilities, academics, administration, departments, faculty, contact us and so on. Some of the university/college has faculty information to inner web pages. Hence, extracting faculty information from these kinds of websites at the same time is impossible. This research mainly focused on extracting the required information the institutional websites. There are four phases in the proposed methodology. They are a web data collection, web data storage, web data process and web information extraction. The web data are collected from various web pages using Depth First Search algorithm and the collected are stored in the ontology. From the ontology the web data are processed using Multimodal Cross Reference (MCR) Re-Ranking, Document Clustering Algorithms, and Automatic Annotation method. The processed web data are again stored in the ontology. Finally the required information is extracted. In order to this proposed model, we applied for extracting required faculty information from educational websites. The proposed model is used to extract faculty information from various Arts and Science universities and colleges and to send relevant information to the faculties concerned. It helps the user to extract faculty details and information is communicated, i.e. sending invitations, conference brochures, workshop information, etc. to those faculty members at the same time.

Keywords: Web Content Extraction, Depth First Search, Ontology Generation, Multimodal Cross Reference Re-ranking, Text-based Web Document Clustering, Automatic Annotation Method

1. INTRODUCTION
Web mining is the process of determining probably useful and previously undiscovered information from the web. It is the integration of information, gathered from World Wide Web, using traditional data mining methodologies and techniques. Web mining is classified into three different kinds: structure mining, usage mining and content mining [14]. Content mining is the extracting significant information from web documents which contain text, audio, HTML pages, images, structured records, video, animation and Metadata. Structure mining is the method of extracting structured data from the web documents. The structure of a graph incorporates with the hyperlinks and web pages, wherein the web pages are considered as nodes. The hyperlinks are considered as edges and are connected to related pages. Usage mining is used to discover interesting usage information from the web. It is used to investigate the performance of online users [11]. Web usage mining is also termed web log mining [17]. It includes the steps such as data collection, pre-processing, information detection, and pattern analysis [5] [18]. Web content mining, commonly engages document tree extraction, web content extraction, classification, and data clustering and labeling of the attributes for results. Web content extraction methods like wrapper generation and semantic web ontology support more tools for extracting relevant information from the web. Web content mining techniques covers classification, summarization and clustering the web contents.

Abstract:

In recent years, a single website has several numbers of web pages which contain structured or/and unstructured information. Hence, different websites contain different web pages which are structured, unstructured, or semi-structured. Extracting relevant information from these websites is not an easy task. Web content mining tools are used and these tools help the user to extract relevant information from the web. The existing tools are open source and also user required to select the information manually. To extract web information from various web pages at the same time is a very complex task because the web returns lots of results and to extract the user required information, it takes an increased amount of browsing effort. To overcome this difficulty developed a tool for extracting the web information. The main objective of this research is to extract the web information from various web pages.

1.1 Motivation
In recent years, a single website has several numbers of web pages which contain structured or/and unstructured information. Hence, different websites contain different web pages which are structured, unstructured, or semi-structured. Extracting relevant information from these websites is not an easy task. Web content mining tools are used and these tools help the user to extract relevant information from the web. The existing tools are open source and also user required to select the information manually. To extract web information from various web pages at the same time is a very complex task because the web returns lots of results and to extract the user required information, it takes an increased amount of browsing effort. To overcome this difficulty developed a tool for extracting the web information. The main objective of this research is to extract the web information from various web pages.

1.2 Contribution
This research mainly focused on extracting the required information the institutional websites. There are four phases in the proposed methodology. They are web data collection, web data storage, web data process and web information extraction. The web data are collected from various web pages using Depth First Search algorithm and the collected are stored in the ontology. From the ontology the web data are processed using Multimodal Cross Reference (MCR) Re-Ranking, Document Clustering Algorithms, and Automatic Annotation method. The processed web data are again stored in the ontology. Finally the required information is extracted. In order to this proposed model, we applied for extracting required faculty information from various Arts and Science universities and colleges and to send relevant information to the faculties concerned. It helps the user to extract faculty details and information is communicated, i.e. sending invitations, conference brochures, workshop information, etc. to those faculty members at the same time.

Contribution:

The proposed model is used to extract faculty information from various Arts and Science universities and colleges and to send relevant information to the faculties concerned. It helps the user to extract faculty details and information is communicated, i.e. sending invitations, conference brochures, workshop information, etc. to those faculty members at the same time.

Author1 is currently working as Assistant Professor, Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu, India. E-mail: elasugan1992@gmail.com

Co-Author1 is currently pursuing Ph.D in Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu, India. E-mail: vijimohan_2000@yahoo.com
information from educational websites. The contributions of this research work as follows,

- This research work proposed a novel and efficient framework for extracting faculty information from the education institutions websites.
- The proposed approach including Multimodal cross reference re-ranking for ranking the web documents and also proposed Text-based Web Document Clustering has been applied and assessed for web document clustering.

The rest of the paper is organized as follows: section 2 explains the various related works and section 3 illustrates the methods of web data collection, web data storage using ontology and web data processing. Section 4 clarifies the observations of the experiments carry out on the dataset, results comparison with existing methods and the implementation. Finally, section 5 discusses the conclusion of this paper and recommends for the future enhancement.

2. RELATED WORK

R. Cooley et al. [21] proposed a structure for web mining using various web mining tasks and implemented a prototype which is named as Webminer. The authors presented two innovative and agile-developing domains: Data Mining and Semantic Web. The author recommended the means by which these domains can be combined. Three distinct approaches for semantic web mining were presented. Yuefeng Li et al. [24] established an ontology mining method for fetching suitable information from the web. They also presented a peculiar method for conquering deriving patterns in order to filter the detected ontology. Manjot Kaur et al. [15] examined web document clustering approaches which differ in several portions like types of attributes used to categorize documents, similarity measures used, depiction of the clusters, etc. Based on the characteristics of the documents are clustered with the various approaches are classified based on text, link, and hybrid. The authors proposed text-based clustering algorithms which mainly focused on the minor similarity based clustering to discover the initial cluster center’s efficiencies and moreover reduces time complexity. Yiyao Lu et al. [25] studied about the annotation process. The process initially aligns the data units on a result page into collections in such a way that the data in the same collection have the identical semantics. The authors used clustering-based shifting method which is capable of managing a diversity of relationships between HTML text nodes and data units containing one-to-nothing, one-to-one, many-to-one and one-to-many. Finally, the authors found the finest technique to solve data alignment problem. Aviral Nigam [1] studied various web crawler algorithms. The author compared the performance of different web crawling algorithms. Fish Search, Breadth First Search, A* Search, Best First Search Adaptive A* Search algorithms were used for analysis. They analysed that the Breadth First Search is also known as blind search algorithm; hence it is not an effective method [1]. Pavalam S M et al. [18] discussed the various searching algorithms like Depth First Search, Breadth First Search, Page Rank Algorithm, Best First Search, Genetic Algorithm and Naïve Bayes Classification Algorithm. The researchers compared and analysed the strengths and weaknesses of these algorithms. Chitra Kalyanasundaram et al. [7] described the document/text clustering using supplementary information and the content of generating cluster with higher purity. The supplementary information may include documenting information, links in a document, index terms used within a document, or any other data that is not generally used for clustering. The authors reviewed the ways by which information can be associated with the data in different applications of text domain which is used for the clustering algorithms like k-means and text clustering. The performance analysis shows that text clustering performs better than k-means clustering algorithm which has been examined by the performance factors like precision, recall, and F-Score. Ameesha Reddy et al. [2] presented a flexible and effective re-ranking method termed Cross-Reference (CR) Re-ranking to enrich the performance of retrieval of videos. Rasoul Dezhkam et al. [19] discussed an automatic ontology construction. They developed a tool which automatically generates the ontology from the existing documents on the web. The authors then, with a standardized assessment procedure, analyzed the efficiency of ontology. Teena Merin Thomas et al. [23] described automatic data extraction from heterogeneous web pages. Many websites holds large collections of pages which are created using typical templates. Templates negatively impact the performance and lead to wastage of resources. Hence, to avoid these problems, the authors proposed RTDM and TEXTMDL approaches for identifying the template and extracting the data from the heterogeneous web pages.

3. METHODS

3.1 Web Data Collection

Web data are collected from websites of different universities and colleges. A single website consists of more than 100 web pages and a web page contains lots of information. Each university/college website has different number of web pages. Some university/college website has faculty information in the same web page and some websites have it in different web pages. Hence the number of web pages of each university/college websites may differ from each other. In this process, Depth First Search algorithm is used for searching and extracting the faculty information from the university/college website. The extracted web data is known as Search Result Records (SRR) [9]. It searches and extracts the information from web pages to the relevant query [26]. Here, we considered details of faculties from five departments such as Tamil, English, Mathematics, Physics and Computer Science from 150 colleges and 7 Universities. The extracted results are stored into the ontology.

### DFS Algorithm

<table>
<thead>
<tr>
<th>Aim: Explore every vertex and every edges in HTML Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: HTML Documents</td>
</tr>
<tr>
<td>Output: Extracted faculty Information</td>
</tr>
<tr>
<td>Let Graph G= (V, E)</td>
</tr>
<tr>
<td>// Depth first search: Recursive algorithm</td>
</tr>
<tr>
<td>DFS(ν= vertex)</td>
</tr>
<tr>
<td>Iteration = adjacent (v)</td>
</tr>
</tbody>
</table>
3.2 Web Data Storage using Ontology

After searching, the next step of this research is to store Search Result Record (SRR) in the ontology. Ontology is well documented and easily understandable architecture of the Semantic Web. In addition, it plays a fundamental role in automated analysis in a domain by declaring the existing entities and the relationship between them. Ontology should be constructed highly with the importance of ontology properties [27]. It has become a significant task to boost the performance of ontology development. The ontology building by manual is a complex task i.e. it consuming long period for ontology building and also the cost is high. It is an efficient way to improve the efficiency of the ontology construction by constructing it from existing database resources. However, the task is very challenging too. The relational schema is mapped to ontology during the analysis of the relations between the primary key and attributes. The relational data are then mapped to ontology instances [22]. In this research, mapping rules among relational database aspects and ontology are proposed based on the existing ontology development method. Finally, the new method based on the relational database is formed to generate ontology automatically [24].

3.3 Web Data Processing

3.3.1 Multimodal Cross Reference Re-Ranking

Ranking the web documents is a significant task for identifying the high ranked web documents from the web search results record. The proposed multimodal cross reference re-ranking method used to cluster the ranked web documents as high, medium and low [6]. It involves three stages: (i) clustering the initial results based on user query (ii) determining the rank of the cluster and (iii) merging the ranked clusters into a new result set using cross reference method. In this method extracted results are processed with two different structures i.e. structure A and structure B. In each of the processes clustering is performed and ranked the clusters as High, Medium and Low according to their relevance to the query [12]. Finally, combined all the ranked clusters from two different structures and formed enhanced result set [14]. This provides high accuracy on high ranked web documents by using cross reference strategy. Here Euclidean distance measure used for finding the smallest distance between the web documents. The high accuracy of the documents are considered as high ranked web documents are extracted using the proposed model. [10] Euclidean distance
\[
md(x_i, X) = \min_{x_j \in X} \{d(x_i, x_j)\}
\]

3.3.2 Web Document Clustering

Clustering algorithms are essential [3] for grouping the faculty members based on their respective departments [3]. Faculty members of the same department from different colleges and universities are discovered from their relevant websites and clustering is performed using the existing and proposed algorithms.

Text-based Web Document Clustering (TBWDC)

The existing text-based clustering algorithm uses only document weight and tf-idf score. Hence, it gives results with low accuracy. To overcome this problem, we have proposed a structure called mathematical matrix and used concept-based similarity measure. This measure improved the accuracy [7]. Similarity between the web documents depends on the concept analysis of sentence, documents, and the corpus levels. The quality of clusters produced is influenced by the similarity measure used it is insensitive to noise while calculating the similarity. This is because the concepts are analyzed in terms of sentence, document and corpus levels and hence, the probability of finding a concept match between unrelated documents is very small. The proposed algorithm gives a better result when compared to the existing text-based clustering algorithm [8]. This proposed enhanced text-based clustering algorithm forms clusters based on similar document. Here, we calculate the weight of the document based on its concepts of the document is calculated. The tfweight, value denotes the weight of concept i in document d at the document level, and the ctfweight, value denotes the weight of the concept i in document d at the sentence level based on the contribution of concept i to the semantics of the sentences in d. The sum of tfweight, and ctfweight, presents an accurate measure of the contribution of each concept to the meaning of the sentences and to the topics covered in the document.
### 3.3.3 Automatic Web data Annotation Method

After clustering, the web documents should be annotated using the annotation method. The annotation process aligns the data units into several collections wherein all data in same collection have the identical semantics. Later, according to grouping, each group annotates it from contrast annotation styles [15]. Labeling is done to give meaningful names to each group of the data units [4]. After data units are labelled successfully, annotation wrapper is automatically constructed for the search sites. This can be used for annotating new result pages from the same Web Database (WDB). Here, the extracted web information is annotated based on the labels. The proposed annotation technique is an enhancement over the existing multi-annotator approach. An automatic annotation method contains three stages: alignment stage, annotation stage, and wrapper generation stage. The first phase is the alignment stage which is to identify all data units in the search records which are extracted from the different web pages and then categorize them into different groups [16]. Each group denotes a concept and each column contain data units of the same concept across all search records.

#### Alignment Stage

The objective of alignment is to move the data units in the table so that every alignment group is well aligned while the order of the data units within every SRR is preserved. In this alignment stage, attributes appear in the same order across all Search Result Records (SRRs) on the same output page though the SRRs which may incorporate with distinct sets of attributes, owing to missing values. This is true in general because the SRRs from the same web database are normally generated by the same template program. Thus, they can conceptually consider the SRRs on a result page in a table format where each row represents one SRR and each cell holds a data unit or is empty if the data unit is not available. Each table column is referred to as an alignment group containing at the most one data unit from each SRR [16]. If an alignment group contains all the data units of one concept and no data unit of other concepts, the group is supposed to be well-aligned.

#### Annotation Stage

This research work has proposed automatic data annotation method. This approach consists of six types of similarities: similarity score, data content similarity, presentation similarity, tag path similarity, data type similarity and adjacency similarity [13].

#### Similarity Score

The purpose of data alignment is to put the data units of the same concept in one group so that they can be annotated holistically. The similarity between two data units, (or two text nodes) \( d_1 \) and \( d_2 \) is a weighted sum of the similarities of the five features between them i.e. [20]

\[
\text{Sim}(\text{wd}_1, \text{wd}_2) = w_1 \times \text{DCS} (\text{wd}_1, \text{wd}_2) + w_2 \times \text{PSS} (\text{wd}_1, \text{wd}_2) + w_3 \times \text{DTS} (\text{wd}_1, \text{wd}_2) + w_4 \times \text{TPS} (\text{wd}_1, \text{wd}_2) + w_5 \times \text{AS} (\text{wd}_1, \text{wd}_2)
\]

(2)

Where \( w_1 \) is the average weight of the two web documents in data content similarity; \( w_2 \) is the average weight of the two web documents in presentation similarity; \( w_3 \) and \( w_4 \) are the average weights of the documents in data type similarity and tag path similarity and \( w_5 \) represents the average weight of adjacency. A commonly used term weighing method is tf-idf. It assigns a higher weight to a term if it occurs frequently in the web document but rarely in the whole document collection. For calculating the tf-idf weight of a term in a particular web document, it is necessary to know two things: The frequency at which it occur in the web document (term frequency = tf) and the number of documents of the collection in which it occurs (document frequency = df). When the the inverse of the document frequency is taken (= idf) both components for calculating the weight are obtained. The weight is calculated by multiplying tf by idf.

\[
idf = \log \left( \frac{N}{df} \right)
\]

(3)

Web Content Similarity (DCS)

\[
\text{DCS}(\text{wd}_1, \text{wd}_2) = \frac{v_{d_1} \cdot v_{d_2}}{||v_{d_1}|| \cdot ||v_{d_2}||}
\]

(4)

---

<table>
<thead>
<tr>
<th>Pseudo code for TBWDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: Select number of clusters ( n ) to be determined</td>
</tr>
<tr>
<td>Step 2: Select initial cluster center as objects ( n ) randomly</td>
</tr>
<tr>
<td>Step 3: Assign each object to its adjacent cluster</td>
</tr>
<tr>
<td>Step 4: Compute mathematical matrix of web document</td>
</tr>
</tbody>
</table>
| Step 5: Next, tf/idf of the term in the web document is calculated. For a term \( i \) in document \( j \):
| \[
| w_{ij} = tf_{ij} \times \log \left( \frac{n}{df_i} \right)
| \]
| Where \( tf_{ij} \) represents the number of occurrences of \( i \) in \( j \), \( df_i \) stands for the number of web documents containing \( i \), \( n \) represents the total number of web documents. |
| Step 6: Calculate the weight of the document based on the concepts of the document. The \( tf_{ij} \) value is normalized by the length of the document vector of the term frequency \( tf_i \) in the document \( d \), where \( j = 1,2,..,c \):
| \[
| \text{tfweight}_i = (\frac{tf_{ij}}{\sqrt{\sum c_{nj}=1 (tf_{ij})^2}})
| \]
| where \( c_{nj} \) is the total number of the concepts which have a term frequency value in the document \( d \). \( \text{ctfweight}_i \) value is normalized by the length of the document vector of the conceptual term frequency \( ctf_{in} \) in the document \( d \) where \( j = 1,2,...,c \):
| \[
| \text{ctfweight}_i = (\frac{ctf_{ij}}{\sqrt{\sum c_{nj}=1 (ctf_{ij})^2}})
| \]
| Step 7: To sum two values of \( \text{tfweight} \) and \( \text{ctfweight} \), \[
| \text{weight} = \text{tfweight}_i + \text{ctfweight}_i
| \]
| Step 8: Compute Similarity measure, i.e. cosine similarity. For two documents \( d_1 \) and \( d_2 \), the similarity between them can be calculated:
| \[
| Sim_c(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1|| \cdot ||d_2||}
| \]
| Step 9: Compute new clusters, i.e. Compute mean points for centroid:
| \[
| c = \frac{1}{|S|} \sum_{d \in S} d
| \]
| where \( S \) is a set of documents; \( d \) is a document and \( c \) is a centroid vector. |
| Step 10: Until no changes on cluster centers (i.e. Centroids do not change location anymore) |
It is the similarity between the term frequency vectors of \( w_{d1} \) and \( w_{d2} \), where \( V_d \) is the frequency vector of the terms inside data unit \( d \); \( \| V_d \| \) is the length of \( V_d \), and the numerator is the inner product of two vectors [16].

**Presentation Style Similarity (PSS)**

It is the average of the style feature scores (SFS) over all six presentation style features such as font style, font size, color of the font, font weight, text decoration, and font face between \( w_{d1} \) and \( w_{d2} \) [20]

\[
PSS(w_{d1},w_{d2}) = \frac{\sum_{i=1}^{6} SFS_i}{6}
\]

Where SFS is the score of the \( i \)-th style feature, and it is well-defined by SFS = 1 if \( F_{i1} = F_{i2} \) and SFS = 0 otherwise, and \( F_{i1} \) is the \( i \)-th style feature of data unit \( d \).

**Data Type Similarity (DTS)**

It is resolved by the typical sequence of the component data types among two data units. The longest common sequence (LCS) cannot be longer than the number of component data types of these two data units. Thus, let \( t_1 \) and \( t_2 \) are taken to be the sequences of the data types of \( d_1 \) and \( d_2 \), respectively. \( TLen(t) \) represents the number of component types of data type \( t \). Then, the data type similarity between data units \( w_{d1} \) and \( w_{d2} \) is [16]

\[
DTS(w_{d1},w_{d2}) = \frac{LCS(t_1,t_2)}{\max(Tlen(t_1),Tlen(t_2))}
\]

**Tag Path Similarity (TPS)**

This is the edit distance (EDT) between the tag paths of two data units. The edit distance here refers to the number of insertions and deletions of tags needed to transform one tag path into the other. It can be seen that the maximum number of possible operations needed is the total number of tags in the two tag paths. If \( p_1 \) and \( p_2 \) are considered to be the tag paths of \( d_1 \) and \( d_2 \) respectively \( PLen(p) \) denotes the number of tags in tag path \( p \), the tag path similarity between \( d_1 \) and \( d_2 \) is [16]

\[
TPS(w_{d1},w_{d2}) = 1 - \frac{EDT(p_1,p_2)}{\max(PLen(p_1) + PLen(p_2))}
\]

**Adjacency Similarity (AS)**

The adjacency similarity between two data units \( d_1 \) and \( d_2 \) is the average of the similarity between \( d_1^p \) and \( d_2^p \), and the similarity between \( d_1^s \) and \( d_2^s \) is

\[
AS(w_{d1},w_{d2}) = \frac{Sim'(w_{d1}^p,w_{d2}^p) + Sim'(w_{d1}^s,w_{d2}^s)}{2}
\]

When computing the similarities (Sim0) between the preceding or succeeding units, only the first four features are used. The weight of adjacency feature \( w_5 \) is proportionally distributed to other four weights [15].

**Wrapper Generation Stage**

Annotation wrapper is a description of the annotation rules for all the attributes in the result page. After the annotation is completed, the wrapper is generated automatically for the annotated result group. The wrapper can be applied to efficiently annotate the SRRs extracted from the same web database with new queries. After performing the automatic annotation method, aligned search results are stored again in the ontology. From the ontology, information is communicated to the recipients. During this process, the proposed system sends the invitation brochures for workshops/conferences and seminars, circulars, meeting information, etc. to the respective faculty members concerned. This communication is essential and useful for both sender and receiver. This system saves time and reduces communication and printing costs.

**4. EXPERIMENTS**

This experiment is conducted using Microsoft Visual Studio on the system with an Intel Core (TM) 2 Duo processor running at 2.20GHz, 2 GB RAM, 32 bit Windows 7 Ultimate. Performance test is evaluated based on search time. In each stage, we have compared the performance of the existing algorithms with that of the proposed algorithms. Different performance metrics apply to each stage. Experimental results are presented in the following section.

**4.1 Dataset**

In order to perform the experiments, the real time dataset is collected from different web pages. This dataset contains seven Arts and Science University/college faculty information. It has 1240 instances and 8 attributes namely university name, department name, faculty name, qualification, designation, phone number, address, email id and URL.

**4.2 Result and Discussion**

The performance of web crawling algorithms is analysed to identify the best web crawling algorithm. It is observed that depth first search algorithm gives better results compare to breadth first search algorithm. Precision is the proportion of relevant documents in the results returned. Recall is the ratio of relevant documents found in the search result to the total of all relevant documents. F-Score is a way of combining recall and precision scores into a single measure of performance. Table 4.1 and figure 4.1 shows the performance measures for web crawling algorithms such as breadth first search and depth first search algorithm.

| Table 4.1 Performance Measure for Web Crawling Algorithms |
|-------------|----------|----------|----------|
| Web Crawling Algorithms | Precision | Recall | F-Score |
| BFS         | 88.8     | 90.4     | 89.59    |
| DFS         | 91.2     | 92.6     | 91.89    |
In this research work, three existing algorithms, i.e. text-based clustering and k-means clustering algorithms and hierarchical agglomerative clustering are compared with proposed Text-based Web Document Clustering (TBWDC) algorithm. Experimental results analysed the performance of these with the help of five performance factors. They are clustered instances, unclustered instances, accuracy, execution time, and iteration. Here we assign tuples into clusters according to their properties. To compute accuracy, each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N.

$$\text{Clustering Accuracy}(\Omega, C) = \frac{1}{N} \sum_{k} \max |\omega_k \cap c_j|$$  \hspace{1cm} (9)

where $\Omega=\{\omega_1, \omega_2, ..., \omega_k\}$ is the set of clusters and $C=\{c_1, c_2, ..., c_l\}$ is the set of classes.

According to the performance analysis shows that Text-based Web Document Clustering (TBWDC) algorithm gives better results than other clustering algorithms. Table 4.2 and Figure 4.2 shows the performance evaluation of clustering algorithms involving clustering instances, unclustered instances, and accuracy. The evaluation shows that TBWDC algorithm gives better result of 93% accuracy compared with other algorithms.

**Table 4.2 Performance Evaluation for Clustering Algorithms**

<table>
<thead>
<tr>
<th>Clustering Algorithms</th>
<th>Clustering Instances (%)</th>
<th>Unclustered Instances (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Based</td>
<td>89</td>
<td>11</td>
<td>89</td>
</tr>
<tr>
<td>K-Means</td>
<td>87</td>
<td>13</td>
<td>87</td>
</tr>
<tr>
<td>HAC</td>
<td>85</td>
<td>15</td>
<td>85</td>
</tr>
<tr>
<td>TBWDC</td>
<td>93</td>
<td>7</td>
<td>93</td>
</tr>
</tbody>
</table>

Table 4.3 and Figure 4.3 and Figure 4.4 shows execution time and the number of iterations needed to form the cluster. The chart illustrates that TBWDC algorithm takes a minimum number of iterations and less time taken to form the cluster.

**Table 4.3 Time Taken and Iteration to Form the Respective Number of Clusters**

<table>
<thead>
<tr>
<th>Clustering Algorithms</th>
<th>Iteration</th>
<th>Time Taken (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Based</td>
<td>18</td>
<td>4313</td>
</tr>
<tr>
<td>K-Means</td>
<td>29</td>
<td>6424</td>
</tr>
<tr>
<td>HAC</td>
<td>26</td>
<td>6156</td>
</tr>
<tr>
<td>TBWDC</td>
<td>7</td>
<td>2116</td>
</tr>
</tbody>
</table>

**Figure 4.1 Performance Measure for Web Crawling Algorithms**

**Figure 4.2 Performance Evaluations for Clustering Algorithms**

**Figure 4.3 Time Taken to Form the Respective Number of Clusters**
The ranked web documents are stored in the ontology. From that, ranked web documents are clustered using clustering algorithms. The similar documents are clustered together. The proposed system is experimentally evaluated based on precision and recall measures which are used for information retrieval, to evaluate the performance of automatic annotation method. For alignment, precision is defined as the percentage of the correctly aligned data units over all the aligned units by the system, and recall is the percentage of the data units that are correctly aligned by the system overall manually aligned data units by the expert. Table 4.3 and figure 4.4 show that the performance of data alignment for search result record.

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{10}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{11}
\]

**Table 4.4 Performance of Data Alignment**

<table>
<thead>
<tr>
<th>Department</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subject 1) Tamil</td>
<td>98.6</td>
<td>98.2</td>
</tr>
<tr>
<td>(Subject 2) English</td>
<td>98.4</td>
<td>97.3</td>
</tr>
<tr>
<td>(Subject 3) Mathematics</td>
<td>99.2</td>
<td>99.5</td>
</tr>
<tr>
<td>(Subject 4) Physics</td>
<td>95.2</td>
<td>100</td>
</tr>
<tr>
<td>(Subject 5) Computer Science</td>
<td>99.0</td>
<td>99.1</td>
</tr>
<tr>
<td>Overall Average</td>
<td>98.8</td>
<td>98.82</td>
</tr>
</tbody>
</table>

**Figure 4.5 Performance of Data Alignment**

**Table 4.5 Performance of Annotation**

<table>
<thead>
<tr>
<th>Department</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subject 1) Tamil</td>
<td>97</td>
<td>96.2</td>
</tr>
<tr>
<td>(Subject 2) English</td>
<td>97.4</td>
<td>96.3</td>
</tr>
<tr>
<td>(Subject 3) Mathematics</td>
<td>95.2</td>
<td>99.5</td>
</tr>
<tr>
<td>(Subject 4) Physics</td>
<td>95.2</td>
<td>97.5</td>
</tr>
<tr>
<td>(Subject 5) Computer Science</td>
<td>97.6</td>
<td>97.1</td>
</tr>
<tr>
<td>Overall Average</td>
<td>96.48</td>
<td>97.32</td>
</tr>
</tbody>
</table>

**Figure 4.6 Performance of Annotation**

**Table 4.6 Performance of Annotation with Wrapper**

<table>
<thead>
<tr>
<th>Department</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Subject 1) Tamil</td>
<td>94.5</td>
<td>91.2</td>
</tr>
<tr>
<td>(Subject 2) English</td>
<td>93.7</td>
<td>92.5</td>
</tr>
<tr>
<td>(Subject 3) Mathematics</td>
<td>95.6</td>
<td>94.2</td>
</tr>
<tr>
<td>(Subject 4) Physics</td>
<td>93.2</td>
<td>94.5</td>
</tr>
<tr>
<td>(Subject 5) Computer Science</td>
<td>95.1</td>
<td>93.8</td>
</tr>
<tr>
<td>Overall Average</td>
<td>94.42</td>
<td>93.24</td>
</tr>
</tbody>
</table>
Figure 4.7 Performance of Annotation with Wrapper

Table 4.5 and figure 4.6 shows the performance of an annotation method for search result record. Table 4.6 and figure 4.7 shows the performance of annotation with a wrapper method for search result record. After clustering, web documents should be annotated using the annotation method. Annotation process aligns the data units into several groups where data in the same group have the same semantics. Then, according to grouping, each group annotates it from different styles and then labels it. Faculty information is annotated based on the university/college and the department.

Table 4.7 Searched and Extracted Information from Websites

<table>
<thead>
<tr>
<th>No. of Departments</th>
<th>No. of University Websites</th>
<th>No. of Colleges Websites</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>Searched Extracted</td>
<td>167</td>
<td>150</td>
</tr>
</tbody>
</table>

Figure 4.8 Searched and Extracted Information

5. CONCLUSION AND FUTURE SCOPE

The purpose of web mining is to discover and extract potentially useful and previously unknown information or knowledge from the web. The main objective of this research is to extract required information from to extract faculty information from various Arts and Science the web pages of educational websites. We applied this proposed model universities and colleges and to send relevant information to the faculties concerned. This research helps the user to extract required information from various web pages using four significant phases: Web Data Collection, Web Data Storage, Web Data Process and Web Information Extraction. In each phase, different techniques, methods, and algorithms are proposed and their results are compared and analysed with the existing algorithms. Nowadays, many web content extraction tools are available, but a major drawback is that the user must enter the URL. This proposed model helps the user to extract faculty details and send information to us. Further, in the future, this research will attempt to collect faculty information from all the universities and the colleges (Arts/Science, Engineering, Medical, Agriculture, Law, etc.) in Tamilnadu. Also, new algorithms will be developed for improving the accuracy for handling large data sets.

6 ACKNOWLEDGEMENT

The authors thank the University Grants Commission (UGC), New Delhi (Official Memorandum No.F1-17.1/2016/RGNF-2015-17-SC-TAM-20541) for the financial support under Rajiv Gandhi National Fellowship for this research work.

REFERENCES

[9]. Dr.S.Vijayarani, Ms.E.Suganya (2016), Web Crawling Algorithms – A Comparative Study,


[12]. HengJi, Juan Liu, Benoit Favre, Dan Gillick, Dilek Hakkani-Tur, Re-ranking Summaries based on Cross-document Information Extraction.


[16]. Mr. Chaitanya Bhosale, Prof. Sunil Rathod, Review on Automatic Annotation of Query Results from Deep Web Database, Multidisciplinary Journal of Research in Engineering and Technology, Volume 2, Issue 1 Pg. 359-364


[22]. Shufeng Zhou, Haiyun Ling, Mei Han (2010), Huaiwei Zhang Ontology Generator from Relational Database Based on Jena, Computer and Information Science, Vol. 3, No. 2.


[25]. Yiyao Lu, Hai He, Hongkun Zhao, Wei Yi Meng (2013), Annotating Search Results From Web Databases, IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 3.
