Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6335201, 10 pages
https://doi.org/10.1155/2022/6335201

*Research Article*

# An Efficient Mechanism for Deep Web Data Extraction Based on Tree-Structured Web Pattern Matching

**B. Bazeer Ahamed,[1] D. Yuvaraj,[2] S. Shitharth,[3] Olfat M. Mirza,[4] Aisha Alsobhi,[5] and Ayman Yafoz[5]**

[1]*Department of IT, University of Technology and Applied Sciences Al Musannah, Oman*
[2]*Department of Computer Science Cihan University—Duhok, Kurdistan Region, Iraq*
[3]*Department of Computer Science and Engineering, Kebri Dehar University, Kebri Dehar, Ethiopia*
[4]*Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia*
[5]*Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

Correspondence should be addressed to S. Shitharth; shitharths@kdu.edu.et

Received 11 April 2022; Accepted 12 May 2022; Published 27 May 2022

Academic Editor: Kuruva Lakshmanna



The World Wide Web comprises of huge web databases where the data are searched using web query interface. Generally, the World Wide Web maintains a set of databases to store several data records. The distinct data records are extracted by the web query interface as per the user requests. The information maintained in the web database is hidden and retrieves deep web content even in dynamic script pages. In recent days, a web page offers a huge amount of structured data and is in need of various web-related latest applications. The challenge lies in extracting complicated structured data from deep web pages. Deep web contents are generally accessed by the web queries, but extracting the structured data from the web database is a complex problem. Moreover, making use of such retrieved information in combined structures needs significant efforts. No further techniques are established to address the complexity in data extraction of deep web data from various web pages. Despite the fact that several ways for deep web data extraction are offered, very few research address template-related issues at the page level. For effective web data extraction with a large number of online pages, a unique representation of page generation using tree-based pattern matches (TBPM) is proposed. The performance of the proposed technique TBPM is compared to that of existing techniques in terms of relativity, precision, recall, and time consumption. The performance metrics such as high relativity is about 17-26% are achieved when compared to FiVaTech approach.

## 1. Introduction

In today's world, a web page contains a large quantity of structured data and requires a variety of web-related apps. Web data is searched in a deep web database using related web query interfaces. Deep web pages, on the whole, contain more accurate and important information and are fetched in response to user requests. The deep or concealed data is the information that has been retrieved. Deep data is presented to users dynamically in the form of HTML documents, along with other material, on web pages in the form of data records. Furthermore, incorporating such retrieved data into integrated structures necessitates significant effort. Because the obtained web pages are only good for eyesight and not for data exchange. As a result, the first and most important step for web information combination is to extract the relevant data on web pages from the exact websites. Absolute web pages share the same template as web pages generated with predefined structure owing to triggered data values, which is a crucial element of web page extraction.

Despite the fact that several ways for deep web data extraction are offered, very few research address template-

related issues at the page level. FiVaTech, which automatically notices the extracted data of a website, is one such existing web page extraction technology in [1]. FiVaTech, on the other hand, is limited to two or three sets of web pages. This paper presented a new page generation representation based on tree-based pattern matches (TBPM) for web data extraction with a larger number of online pages. TBPM's main goal is to create the schema and templates for each particular deep website, which may contain a singleton or several data entries in a single web page. The performance of the TBPM approach with a vast set of databases for deep web data extraction is compared to existing FiVaTech in an experimental evaluation. Metrics including relativity, number of schemas, and execution time are used to evaluate the TBPM scheme's ability to reach a higher percentage in terms of relative data extraction over a set of web pages.

Web data extraction systems are web-related applications functioning in the process of information extraction from web sources such as web pages. A web information extraction system typically communicates with a web page and retrieves data stacked in database. Additionally translate the extracted data in the most interesting structure forms loading in the page for further usability. As known World Wide Web is the biggest database, most relevantly comprise of data facilitating the user needs. The idea and act of web data extraction process are believed from a different point of view [2]. Moreover, the extraction process controls on logical tools impending from different orders like machine learning, logic, and natural language processing. In web data extraction process, many aspects are considered for retrieving data. Some of the aspects are self-determining of the particular application domain where the web data extraction is performed. Other aspects as an alternative closely rely on the specific properties of the application domain.

The major challenges meet in the formulation of a web data extraction process are illustrated as follows:

(i) Web data extraction approaches are always in need of help from human experts. The foremost challenge is to offer a high scale of computerization by minimizing human experts as much as possible. Human comments though play a vital role in increasing the rate of accuracy achieved by a web data extraction approach. An appropriate challenge is consequently to determine a sensible trade-off between the requirements of developing highly automated web data extraction concepts and the needs of achieving extremely accurate performance

(ii) Web data extraction approaches should be able to process a large amount of data in reasonably short span of time. More specific requirements are key vital in the field of business and competitive intelligence, as a company requires performing timely analysis of market criteria

(iii) Applications related to web need additional promising security measures more particularly for applications handling vibration a related data. Hence, intentionally or unintentionally intruder tries to breach user privacy should be appropriately or sufficiently identified and submitted. The intruder actions also cause data leakage. Even though data leakage detection in [3] is able to trigger data leaks through overlapping, the extraction process probably results in collision or fake tuple identification

(iv) Approaches depending on machine learning frequently need an extensively large training set of manually labeled web pages. Typically, the process of labeling pages is time-expensive and error-prone. Hence, in many situations, consider the occurrence of labeled pages. Similarly, the ontology data extraction (ODE) in [4] labels attributes in the query interfaces for data extraction but misses some attributes in labeling resulting in an inefficient data extraction consuming more time

(v) In a certain scenario, a web data extraction tool needs to consistently extract data from a web database consuming more time. Web page in web databases is always repetitive as in [5], and structural changes happen with no sign thus are unpredictable. Finally, in real-world cases, web extraction approaches develop the need of managing the systems that possibly trigger proper working due to deficiency of flexibility in the presence of repetitiveness and face structural modifications of related web sources

Moreover, the machine learning-based algorithms in the process of data extraction faces some limitations. Not often machine learning-based classified document follows the similar type of classification in the future leading to unknown modification. In addition, human experts are in need to classify documents for the training set. And also the range and type of document that should be in the training set are complicated. Hence, proposed TBPM approach intends in development of potential data extraction technique in tree matching algorithm as machine learning algorithms faces certain demerits. The motivation of this research is to enhance the process of deep web extraction in multidata region using web page extraction; this will help to increase the processing speed and minimize time, to deduce the schema and template for each and individual deep web site using TBTM.

## 2. Related Work

Web data extraction has played an important role in a number of web data analysis requests. The data extraction crisis was designed by the author in [6] as a decoding technique of page generation supported by structured data and tree templates. The author developed an unsupervised, page-level data extraction technique for determining the schema and templates for any website with a singleton or multiple data records in a single webpage.

Web data extraction has played an important role in a number of web data analysis requests. The data extraction crisis was designed by the author in as a decoding technique of page generation supported by structured data and tree templates [7]. The author developed an unsupervised,

page-level data extraction technique for determining the schema and templates for any website with a singleton or multiple data records in a single webpage. It is used to quickly identify the data area and the maximum number of data entries in the same set of pages The Semantic Web, or eventual modifications to the web, must enhance the web with semantic page annotations to support knowledge-level inquiry and analysis. Manually creating these ontologies, on the other hand, is a time-consuming and difficult undertaking. The author of [8] described an automatic extraction strategy for studying domain ontologies for semantic web from the deep web.

To improve the precision of attribute extraction, an approach is proposed to heuristically extract the attributes. During the mining process, characteristics are supplemented with ontology, which is used to get deep semantic knowledge via query interfaces [9]. The inclusion of deep web sources requires object matching [10]. Existing approaches assume that record extraction and attribute segmentation are highly accurate. However, due to the limitations of extraction methods, information arose during the methodology's incomplete procedures. The author will not be able to achieve acceptable results if objects are matched based on noisy and incomplete data. As additional data sources emerge as online databases, hidden behind query forms, the deep web is being shaped. Recognizing this innovative method for data delivery is critical for solving novel data integration difficulties. In [11], the author suggested two ways for obtaining an approximately absolute position of output plan features from a deep web data source: the case form approach and the combination model approach, respectively. The author suggested a technique supported by the location of DIV to pull out main text from the body of web pages by renovating, lingering atomic DIV, and assessing DIV position [12] for the trendy DIV page design in web pages. The visualization of web page content is useful for information extraction, as it avoids the use of sophisticated natural language processing knowledge. The information [13], which was shared in the natural language processing expertise with vision disposition of HTML page in the request of information drawing out for web page, was processed out of relevant research [14].

In response to the current composite implementation crises, high mistake rate, and slow extraction speed of web information extraction expertise, the paper [15] suggested a revolutionary web extraction technique based on the uniqueness of web page construction. To improve the efficiency of information extraction, we need to look at the automatic web information extraction technique [16]. Page clustering is used to find high comparison clusters by supporting the DOM extraction technique, and the accuracy of clustering findings is improved by using the similarity measure [17]. However, the data extraction processes do not meet the accuracy of the results.

## 3. Web Data Extraction Using Tree-Based Template Matching

The major goal of the tree-based pattern match (TBPM) approach is to extract deep web data while taking into account numerous input pages for total process success. The TBPM method for deep web data extraction is broken down into four stages. The procedure of identifying a similar collection of templates from a group of web pages is described in the first step. The process of retrieving data records from websites is described in the second phase. The presence of web page duplication from absolute online sites is discovered in the third phase. Finally, the fourth phase describes how relative information from web pages is grouped. Below is a diagram of the architecture of deep web data extraction based on tree-structured web template matching is shown in Figure 1.

Using the data object model (DOM) trees, the presence of schema is determined. Use DOM trees for all of the retrieved web pages in the same way. In addition, you can join all of the DOM trees into a single tree. Leaf nodes are recognized from the combined set of tree to find repeating nodes. The resulting tree is then used to distinguish between the website's template and schema.

Consider combining DOM trees with almost the same root node and attempting to build the tree from the bottom up. By specifying the child nodes for each tree, create a matrix representation for the created tree. The system then goes through four critical processes, which are detailed in the sections below: tree creation with templates, matrix representation based on tree, pattern mining, and optional node merging [18].

*3.1. Tree Construction with Templates.* The tree structure makes it easier to determine whether a website has a similar set of templates. The trees are constructed to see if two subtrees with the same root tags are related or not. Identify the similarity match of leaf nodes to determine the subtree matching performance. Instead of replacing the entire root of the subtree, modify the child leaf nodes based on the node replacement technique once the leaf nodes in the subtrees are matched. On the contrary, if the leaf node does not match, dynamic programming can be used to replace the nodes by matching leaf nodes. Meanwhile, the matching is supported with set-type data and is regularized from 0 to 1. The sample tree representation to address the elements of web pages is shown in Figure 2.

The data are typically indicated in tagged sequenced rooted trees, where tags represent the HTML mark-up language syntax. The tree structure indicates the various stages of building of elements forming the web page. The indication of a webpage by using tagged sequenced rooted trees is generally referred as data object model (DOM). HTML tags are built in hierarchical tree manner, one into another in the DOM with leaf nodes as HTML tags. The tree construction facilitates the process of matrix representation.

*3.2. Matrix Representation Based on Tree.* After constructing a tree, a matrix representation is created by specifying the nodes in the matrix $M$. As a result, if the matching score of two nodes with the same tag is greater than a threshold $T$, the matching sign is used to identify them. When two text nodes in the tree representation have similar text values, they get the same symbol; otherwise, they get a different set of symbols. The matrix alignment algorithm aligns the nodes on the

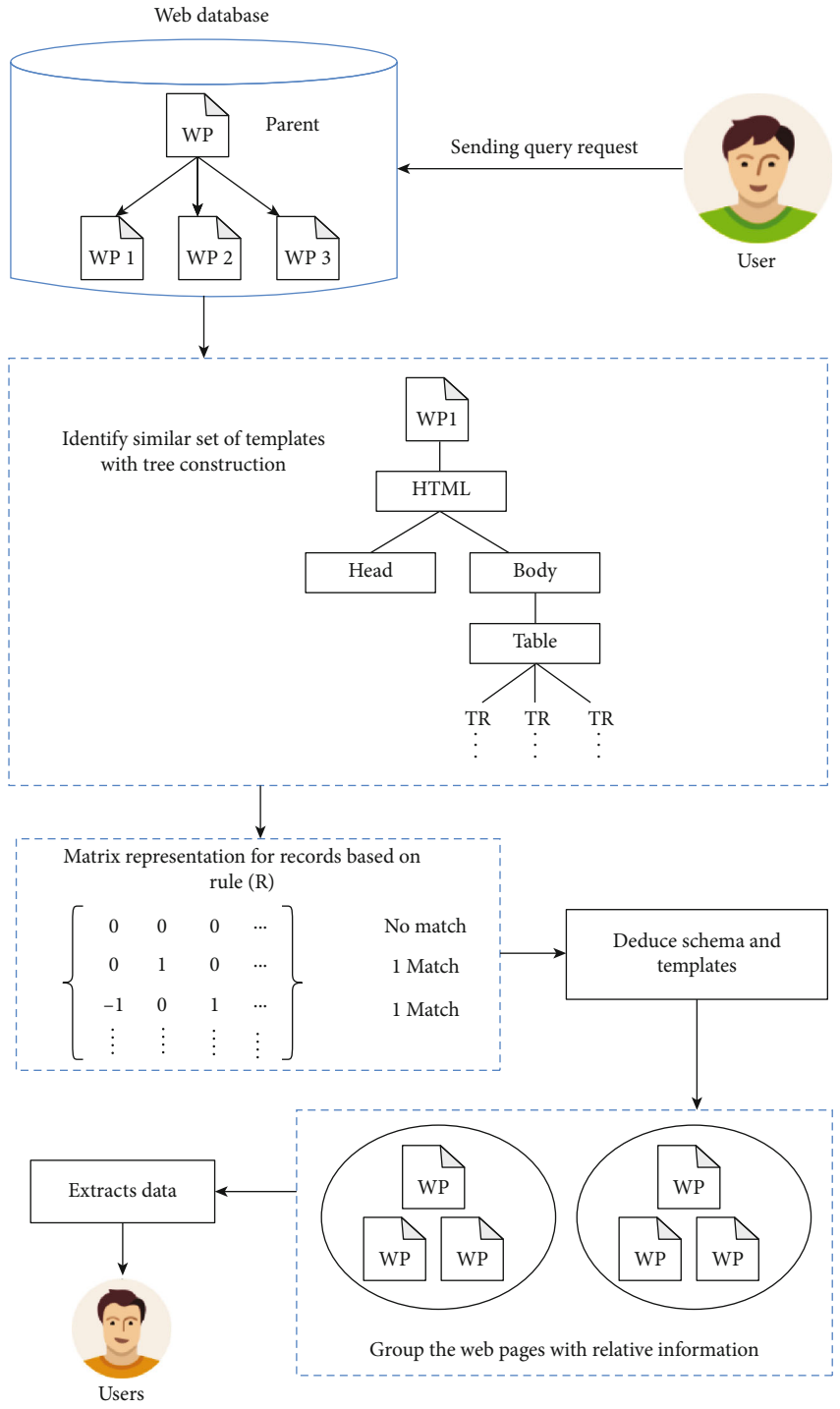


Figure 1: Architecture diagram of deep web data extraction based on tree-structured web template matching.

matrix $M$ row by row in a parallel way. The parallel alignment facilitates the conversion of $M$ into an associated matrix in which each row contains either the same symbol for each column or leaf nodes of a diverse set of symbols designated as basic-typed. The user lists the set of nodes from the related matrix $M$, where each node is described in a defined row.

The task was aligned in each row of the created matrix. Row determines whether or not the row is related to a similar symbol. If a row is associated with a similar symbol or if the child leaf nodes of various symbols occur only in its existing column, the nodes are designated as deviations. If a row is discovered to be unrelated to other symbols, the matrix alignment procedure is used to make the row parallel. Every iteration, a column or node called ShiftColumn is picked from the current row, and all the nodes in that column are moved downstairs in the matrix $M$ by shiftLength. In addition, use an illogical node to scrape the blank spaces. The matrix alignment process is demonstrated in the below steps.

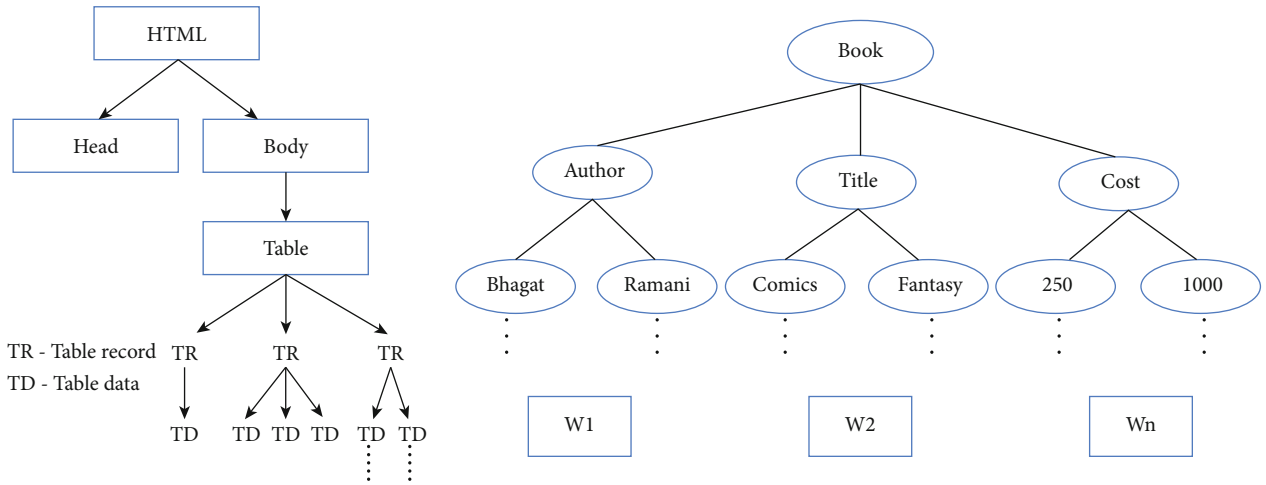


Figure 2: Sample tree representation to address elements of web pages.

Step 1: Begin.
Step 2: Assign row =1 and shiftLength =0 for Matrix M.
Step 3: If (M! = align) then.
Step 4: Align alignedRow (row, M).
Step 5: ShiftColumn (row, shiftLength, 1.
Step 6: M = (row, ShiftColumn, shiftLength).
Step 7: Return M.
Step 8: Else.
Step 9: Increment Row (row = row +1).
Step 10: Go to step.
Step 11: End If.
Step 12: Assign Child leaf node with alignment result (M).
Step 13: End.

Algorithm 1: Matrix alignment procedure.

*3.3. Mining Repetitive Patterns.* When a variety of web pages [19] are provided as input, the procedure of storing set-typed data is straightforward. It is simple to determine the repetitiveness of websites that are also offered as input using a basic technique. However, there are numerous patterns that have yet to be discovered. If a comparable pattern is established in another pattern, the template's assumptions are integrated, resulting in repeating patterns. In the matrix representation step, the influence of missing pattern possibility data is grasped. Data overlapping may occur as a result of merged pattern mining. As a result, TBPM combined the data structure with repetitive pattern mining. Detect each successive tandem repeat or repeating pattern. Also, by erasing all incidences but the first, combine recurring patterns from minute to outsized lengths. This is due to the fact that the predefined data defined here is nested, and each occurrence of a set type should occur simultaneously.

*3.4. Analyzing the Web Pages as Input.* Following recurrent pattern mining, the attention shifts to detecting the presence of choices shown to users based on their queries. The vector of different nodes is used for occurrence.

The occurrence vector for given input web pages$WP = (wpi, wp2, \cdots, wm)$is specified as

$$\text{vector}(wpi, wp2, \cdots.wpi) \quad \text{where wpi} = \begin{cases} 1 & \text{if WP = ith occurrence,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here, wp1 is 1 if WP happens in the $i$th occurrence or 0 otherwise. If WP is not part of a set type, $i$ is the number of input web pages. If WP is part of a set type, then, $i$ sums up of all repetitiveness in all pages.

*3.5. Detection of Data Schemas.* To protect all crucial leaf nodes, the schema of the input web pages should be inferred easily by deleting nodes containing a single child and deficient in types during tree formation. The site presences are indicated in leaf node $k$, and the order of web pages is sorted in the form of a tree [20]. Specify as a tuple if an inner node is not an important node and is not labeled as a set type. By avoiding bogus tuple identification or collision in data extraction, the tuple definition overcomes the constraint of [21]. As seen in the algorithm below, the steps required in the full TBPM technique are as follows.

## 4. Experimental Evaluation Result Analysis

To estimate the performance of the tree-based template matching (TBPM) approach, an experimental evaluation is conducted. On a Pentium 4 1.9 GHz, 512 MB PC, the proposed TBPM method trials are carried out. A large set of web databases is employed, with over 10,000 web database entries. There are numerous domains in which web databases can be found. A variety of users submit query requests in order to obtain relevant data from the web database. Present three sets of user queries for each web database, and aggregate deep web pages with data records at any rate. The TBPM is integrated to the database with the set of web databases to extract the web pages directly from a database on the internet based on user query-related

Step 1: Begin with the set of input web pages you have gathered.
Step 2: Create a tree T.
Step 3: Fill in the missing for each (Tree representation T).
Step 4: Create a M matrix.
Step 5: Determined by the presence of content similarity in web pages,
Step 6: Move the nodes from the left to the right or likewise.
Step 7: End For.
Step 8: Form a set of rules R.
Step 9: For Each (R).
Step 10: Form a vector v in (WP).
Step 11: Determine whether or not there is a schema present.
Step 12: Come to an end For.
Step 13: Make a distinction between the exact representation and the leaf child nodes.
Step 14: Come to an end.
The technique above outlines the full process of identifying schema and templates in order to improve deep web page extraction.

Algorithm 2: Web database WD, data records DR, data items DI, and users U are the inputs.

Table 1: No. of schemas vs. relativity.

| No. of schematics | Relativity (%) FiVaTech | TBPM |
|---|---|---|
| 10 | 18 | 22 |
| 20 | 21 | 26 |
| 30 | 28 | 34 |
| 40 | 34 | 42 |
| 50 | 42 | 53 |
| 60 | 48 | 59 |
| 70 | 52 | 71 |
| 80 | 66 | 79 |

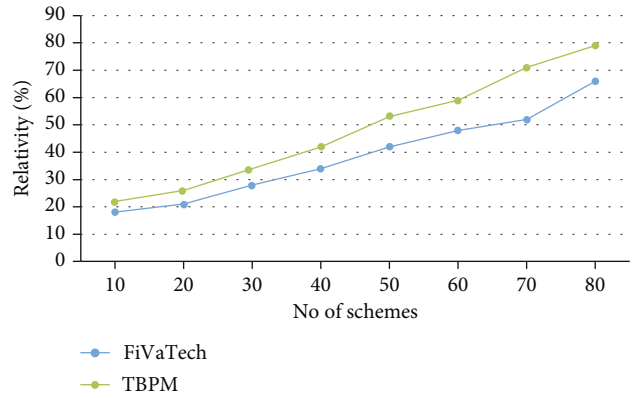


Figure 3: No. of schemas vs. relativity.

information; the data records and their relevant data items are fetched from a set of web pages.

With template matching, the TBPM technique quickly retrieves web pages straight from the web database contained in the multidata region. Data records requested by users are extracted and processed with data items for a variety of web pages. The comparable information is grouped in web pages, allowing the exact data extraction, using a similar set of template-based data records extraction and repeating pattern mining. The users' query-related information is processed after the data records have been extracted. The performance of the TBPM approach and the present FiVaTech approach is described in the table and graph below.

Relativity is defined as the similarity measure between the schema and templates deduction in the web pages and the extracted data contents. More relevant template and schema-based information extraction get higher relativity. The below table depicts the relativity rate of both FiVaTech in [22] and TBPM.

The analysis of the extracted information from the deep web pages is shown in Table 1. The procedure is also depicted based on the number of schemas found in the retrieved templates. The TBPM's value is compared to that of the present FiVaTech technique [4].

The information relativity obtained with the set of schemas on the websites is depicted in Figure 3. In comparison to FiVaTech [5], the TBPM has a high relativity of 17-26% in retrieving exact data from deep web pages. Because TBPM organizes web pages into a tree, with site presences provided in the leaf node, providing the more simple and accurate deduction of scheme as well as template with high relativity. Even though FiVaTech [3] constructs DOM trees with string alignments at each internal node, further clarification of sites is not possible. Precision is the percentage of accurately extracted data form in over all the relevant data forms. Precision retrieves valid data object from the matching website. The higher rate of matching relevant data as per requested query, in addition to span multiple product types, provides higher precision rate.

$$\text{Precision} = \frac{|\text{Pages Predicted as relevant data}| \cap |\text{Pages with relevant data}|}{|\text{Pages Predicted as relevant data}|}. \quad (2)$$

Recall is the percentage of accurately extracted data forms in over all the irrelevant data forms. Recall is also

Table 2: No. of accesses in database vs. precision and recall.

| No. of accesses in web database | FiVaTech | | TBPM | |
|---|---|---|---|---|
| | Precision (%) | Recall (%) | Precision (%) | Recall (%) |
| 100 | 94.3 | 95.3 | 96 | 97.3 |
| 200 | 95 | 95.4 | 96.1 | 97.6 |
| 300 | 95.3 | 95.7 | 96.7 | 98.1 |
| 400 | 95.6 | 96.1 | 97.2 | 98.3 |
| 500 | 95.9 | 96.3 | 97.6 | 98.7 |
| 600 | 96 | 96.8 | 98.1 | 99.1 |
| 700 | 96.5 | 96.8 | 98.3 | 99.3 |
| 800 | 96.7 | 97.1 | 98.4 | 99.8 |

defined as the number of relativity used to predict the relevance between the query and the site.

$$\text{Recall} = \frac{|\text{Pages Predicted as relevant}| \cap |\text{Pages with irrelevant data}|}{|\text{Pages with irrelevant data}|}. \quad (3)$$

For perfect web data extraction approaches, the precision and recall should be high. The below table depicts the both precision as well as recall rate of FiVaTech and TBPM.

Based on the above Table 2 for number of entries in database with respect to precision rate, a graph is depicted below.

The precision rate of the retrieved web pages is depicted in Figure 4 based on the number of entries in the online database. The precision rate of TBPM is about 1-2% high compared with FiVaTech [23]. As TBPM discovers the requested data by pattern matching with repetitive pattern mining, thus, uncategorized information is left with the title extracted from the website. This gives greater precision results and eliminates unwanted information. But FiVaTech adopts node score to match the data which depends on template value. The static page without template struggles in the node score estimation for match detection providing minimum precision rate.

The recall rate of the retrieved web pages is depicted in Figure 5 based on the number of entries in the online database. The recall rate of TBPM is about 2-3% high compared with FiVaTech [23]. As TBPM discovers the repetitive patterns during the pattern mining stage, the left or missing patterns are also spotted in the matrix representation leading to data extraction even in irrelevant identification forms. However, FiVaTech in unable to predict the missing patterns in the prior stage as template values are calculated in later proceedings. Time consumption is defined as the time period required in extracting the data from deep web database.

$$\text{Time Consumption} = \text{Time taken}(T + M + V(\text{wp}) + \text{Data Extraction}). \quad (4)$$

The sum of time taken in tree construction ($T$), matrix representation ($M$), vector occurrence of web pages $V$ (WP),

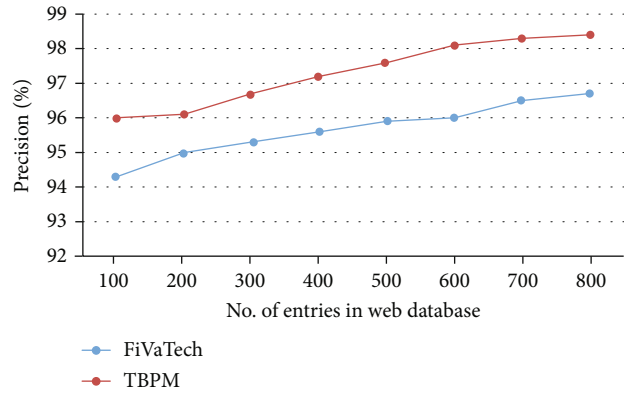


Figure 4: No. of entries in database vs. precision.

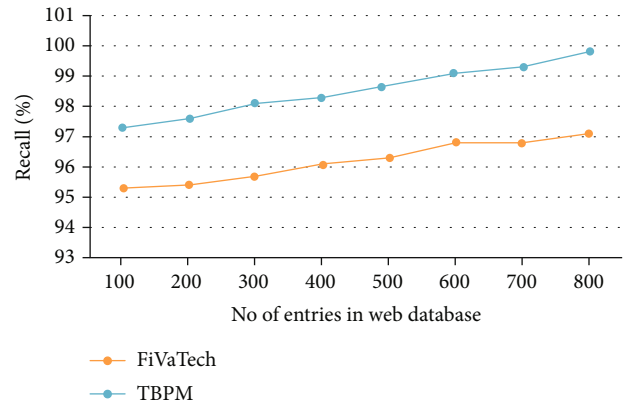


Figure 5: No. of entries in database vs. recall.

and finally, data extraction determines the time consumption. Time usage is calculated in milliseconds and is shown in the Table 3 below.

The time required to extract the relevant deep web pages based on the entries of the web database is illustrated in above Table 3. The consumed time of the TBPM approach is compared with the existing FiVaTech approach.

The time necessary to extract the relevant deep web pages based on the web database entries is depicted in Figure 6. When compared to the conventional FiVaTech approach, the TBPM approach takes 13-17 percent less time to extract web pages. Because TBPM's matrix alignment technique aligns the set of nodes on the matrix $M$ in a parallel row by row fashion, leaf nodes can be reached quickly and without waiting. However, FiVaTech requires a top-down strategy to discover websites in leaf nodes, which takes longer. Finally, using tree construction, matrix representation, pattern mining, and schema identification, the TBPM demonstrated the effective extraction of user-requested information from the web database based on their query. The performance metrics such as high relativity of about 17-26%, precision of 1-2%, recall of 2-3%, and minimum time consumption of about 13-17% are achieved compared to FiVaTech [2] approach. Figures 7–9 compare the precision, recall, and $f$-measure of both existing [22] and propose web data extraction models, respectively. Typically, the

Table 3: Number of database entries vs. time consumption.

| No. of entries in database | Time consumption (ms) | |
|---|---|---|
| | FiVaTech | TBPM |
| 100 | 20 | 15 |
| 200 | 26 | 19 |
| 300 | 31 | 26 |
| 400 | 36 | 30 |
| 500 | 43 | 34 |

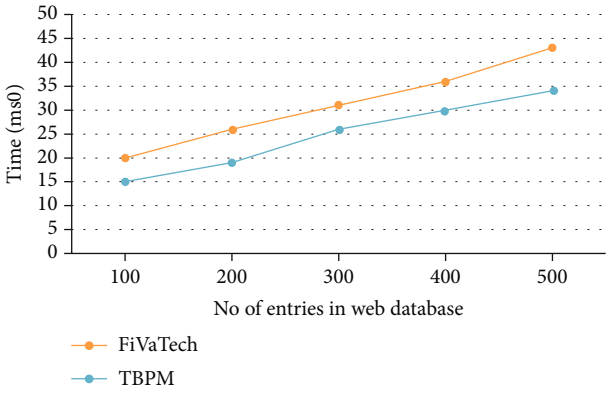


Figure 6: No. of entries in database vs. time consumption.

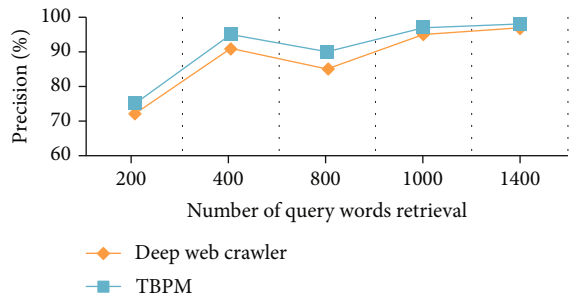


Figure 7: Precision vs. number of query words retrieval.

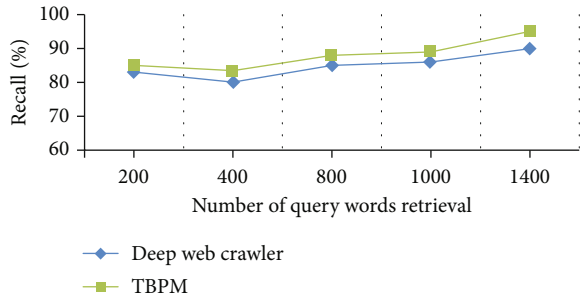


Figure 8: Recall vs. number of query words retrieval.

overall performance and efficiency of the web page data exaction model are determined based on the parameters of precision, recall, and $f$-measure. From the evaluation, it is stated that the proposed TBPM technique outperforms the other approaches with increased values of these measures.

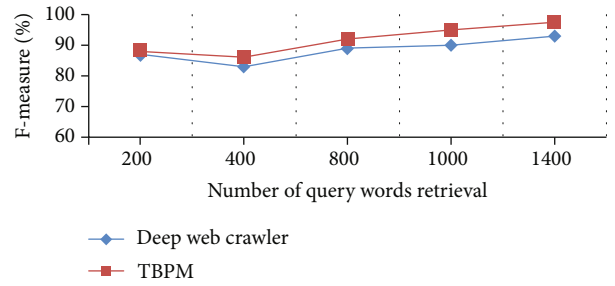


Figure 9: $F$-measure vs. number of query words retrieval.

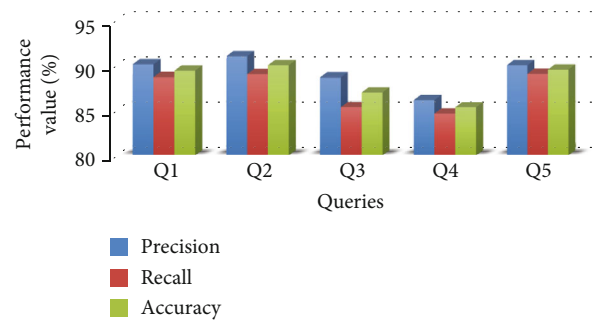


Figure 10: Precision, recall, and accuracy of existing Onto Disco algorithm.

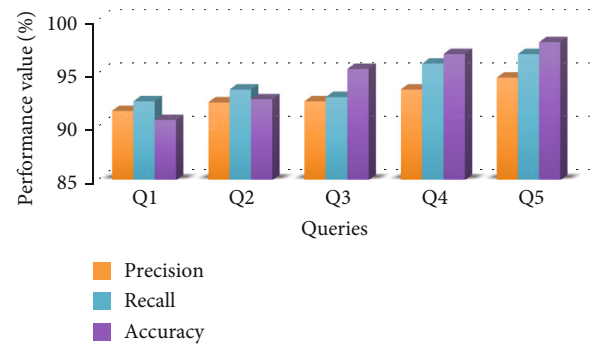


Figure 11: Precision, recall, and accuracy of proposed TBPM.

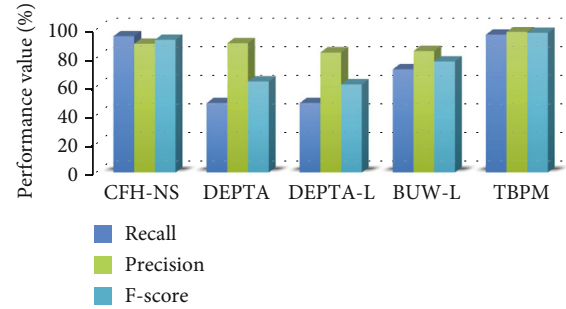


Figure 12: Comparative analysis between existing and proposed techniques.

Figures 10 and 11 validate the precision, recall, and accuracy of existing OntoDisco [22] and proposed TBPM techniques under varying queries. The obtained results indicate that the proposed TBPM technique outperforms the other technique increased precision, recall, and accuracy values

of these measures. Similar to that, Figure 12 compares the existing [24] and proposed data extraction models based on the measures of precision, recall, and $f$-measure. Based on the evaluation, it is observed that the proposed TBPM overtakes the other approaches with increased values of these measures.

## 5. Conclusion

The suggested tree-based pattern matching (TBPM) approach for deep web data extraction blends multiple DOM trees simultaneously. For multiple string alignment, a new technique is being developed that takes optional and set-type data into account. The schema and template for the input website are deduced using the fixed or variant pattern tree and matrix representation. The TBPM matrix alignment algorithm aligns the set of nodes on the matrix $M$ in a parallel row-by-row fashion, allowing leaf nodes to be accessed quickly and without waiting. The lack of flexibility in web databases is avoided by the recurring pattern recognition in pattern mining based on matrix representation. The most significant benefit of the TBPM method is that deep web pages are pulled straight from the database, allowing for a larger number of web pages to be supplied. When compared to existing deep web data extraction technologies as FiVaTech, the TBPM approach outperformed them in terms of high relativity, precision, and recall rate, as well as little time consumption. The performance metrics of the proposed work improved efficiency with high rescission rate of about 1-11%, recall rate of 2-12%, and minimum time consumption of 13-17% compared to the existing approaches, respectively.

## Data Availability

The data that support the findings of the study are available from the corresponding author, upon reasonable request.

## Conflicts of Interest

The authors share no potential conflicts of interest, such as financial interests, affiliations, or personal interests or beliefs, that could be perceived to affect the objectivity or neutrality of the manuscript.

## References


[1] B. B. Ahamed and T. Ramkumar, "An intelligent web search framework for performing efficient retrieval of data," *Computers & Electrical Engineering*, vol. 56, pp. 289–299, 2016.

[2] V. Jose, V. P. Jagathy Raj, and S. K. George, "Ontology-based information extraction framework for academic knowledge repository," *In Proceedings of fifth international congress on information and communication technology*, 2021, pp. 73–80, Springer, Singapore, 2021.

[3] A. Ahmed, A. R. Javed, Z. Jalil, G. Srivastava, and T. R. Gadekallu, "Privacy of web browsers: a challenge in digital forensics," in *In International Conference on Genetic and Evolutionary Computing*, pp. 493–504, Springer, Singapore, 2022.

[4] H. Alani, S. Kim, D. E. Millard et al., "Automatic ontology-based knowledge extraction from web documents," *IEEE Intelligent Systems*, vol. 18, no. 1, pp. 14–21, 2003.

[5] D. Dalal and A. Panwar, "Deep web query extraction algorithm for information retrieval system," *(IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6867–6870, 2014.

[6] S. Deshmukh, P. P. Karde, and V. R. Thakare, "An improved approach for deep web data extraction," in *In ITM Web of Conferences*, vol. 40, p. 3045, EDP Sciences, 2021.

[7] S. Yang and J. Guo, "Improved strategies of relation extraction based on graph convolutional model on tree structure for web information processing," *Journal of Industrial Information Integration*, vol. 25, article 100301, 2022.

[8] C. Jou, "Schema extraction for deep web query interfaces using heuristics rules," *Information Systems Frontiers*, vol. 21, no. 1, pp. 163–174, 2019.

[9] N. Katla, G. Kumar, P. R. Raj, and S. Shitharth, ""Palisade"-a student friendly social media website," in *In Intelligent Computing and Networking*, Springer, Singapore, 2020.

[10] E. Kubias, S. Schenk, S. Staab, and J. Z. Pan, "OWL SAIQL-an OWL DL query language for ontology extraction," in *Proc. of OWLED-07*, CEUR-WS.org, 2007.

[11] B. Liu and J. Xiang, "Au extraction and management of meta information on the domain-oriented deep web," in *In 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 787–790, Beijing, China, 2016.

[12] J. Meneghello, N. Thompson, K. Lee, K. W. Wong, and B. Abu-Salih, "Unlocking social media and user generated content as a data source for knowledge management," *International Journal of Knowledge Management (IJKM)*, vol. 16, no. 1, pp. 101–122, 2020.

[13] M. Padmaja, S. Shitharth, K. Prasuna, A. Chaturvedi, P. R. Kshirsagar, and A. Vani, "Grow of artificial intelligence to challenge security in IoT application," *Wireless Personal Communications*, vol. 119, pp. 1–17, 2021.

[14] H. Xu, S. Jiang, and C. Zheng, "Analysis of ontology semantic tagging method for semantic web-oriented big data," in *In 2021 international wireless communications and Mobile computing (IWCMC)*, pp. 1147–1150, Harbin City, China, 2021.

[15] C. N. Pushpa, G. Deepak, A. Kumar, J. Thriveni, and K. R. Venugopal, "Onto Disco: improving web service discovery by hybridization of ontology focused concept clustering and interface semantics," in *In 2020 IEEE international conference on electronics, computing and communication technologies (CONECCT)*, pp. 1–5, Bangalore, India, 2020.

[16] R. Ranjan, H. Vathsala, and S. G. Koolagudi, "Profile generation from web sources: an information extraction system," *Social Network Analysis and Mining*, vol. 12, no. 1, pp. 1–12, 2022.

[17] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna et al., "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020.

[18] K. Sahu and D. Kelkar, "Enhancing information retrieval by integration invisible web data source," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 4, 2012.

[19] R. Vijay and K. Prasadh, "Tree structured web template matching for deep web extraction," *International Journal of Science, Engineering and Computer Technology*, vol. 3, no. 12, p. 460, 2013.

[20] N. Katla, M. Goutham Kumar, R. Pidugu, and S. Shitharth, "Palisade—a student friendly social media website," in *In Intelligent Computing and Networking*, pp. 53–62, Springer, Singapore, 2022.

[21] Y. Saissi, A. Zellou, and A. Idri, "Extraction of relational schema from deep web sources: a form driven approach," in *In 2014 Second World Conference on Complex Systems (WCCS)*, pp. 178–182, Agadir, Morocco, 2014.

[22] H. A. Santoso, S. C. Haw, and Z. T. Abdul-Mehdi, "Ontology extraction from relational database: concept hierarchy as background knowledge," *Knowledge-Based Systems*, vol. 24, no. 3, pp. 457–464, 2011.

[23] G. K. Gangadhar and A. Kulkarni, "Extraction of product specifications from the web-going beyond tables and lists," in *In 5th joint international conference on Data Science & Management of data (9th ACM IKDD CODS and 27th COMAD)*, pp. 19–27, New York, 2022.

[24] K. Selvakumar and L. Sairamesh, "User query-based automatic text summarization of web documents using ontology," in *In International Conference on Communication, Computing and Electronics Systems*, pp. 593–599, Springer, Singapore, 2021.