XML CLUSTERING FRAMEWORK BASED ON DOCUMENT CONTENT AND STRUCTURE IN A HETEROGENEOUS DIGITAL LIBRARY

Nafisse Samadi*[1]*, Sri Devi Ravana*[2]**

[1,2]Department of Information Systems, Faculty of Computer Science and Information Technology,
University Malaya, 50603 Kuala Lumpur, Malaysia

E-mail: nafissesamadi@gmail.com[1], sdevi@um.edu.my[2*] (corresponding author)

DOI: https://doi.org/10.22452/mjcs.vol36no2.2

ABSTRACT

As textually published information is increasing in digital libraries, efficient retrieval methods are required. Textual documents in a digital library are available in various structures and contents. It is possible to represent these documents with hierarchical levels of granularity when these are organized in XML structure to improve precision by focused retrieval. By this means, contextual elements of each document can be retrieved from a known structure. One solution for retrieving these elements is clustering from a combination of Content and Structural similarities. To achieve this, a novel two-level clustering framework based on Content and Structure is proposed. The framework decomposes a document into meaningful structural units and analyzes all its rich text in its own structure. The quality of the proposed framework was experimented on a heterogeneous XML document collection, having varieties of data sources, structures, and content, be represented as a sample of a real digital library. This collection was made with capabilities to test all of our objectives. The clustering results were evaluated by the Entropy criterion. Finally, the Content and Structure clustering was compared with the usual clustering based on the Content Only to prove the efficacy of considering structural features against the existing Content Only methods in the retrieval process. The total Entropy results of the two-level Content and Structural clustering are almost twice better than the Content Only clustering approach. Consequently, the proposed framework has the ability to improve Information Retrieval systems from two points of view: i) considering the structural aspect of text-rich documents in the retrieval process, and ii) replacing the document-level retrieval with the element-level retrieval.

Keywords: Information retrieval; Document clustering; Focused retrieval; XML document clustering; Digital library

1.0 INTRODUCTION

With the development of digital libraries (DL) and ever-increasing kinds of digital documents, users face a considerable amount of text documents. These documents should be analyzed and organized to be easily accessible and retrievable based on the users' requirements. Most old-style Information Retrieval (IR) systems are dependent on the Vector Space Model (VSM) or the Boolean model to signify the documents' flat structure as a bag of words [1]. These models were then extended as the knowledge-aware and fuzzy Boolean models. However, the structural organization of text documents is ignored in all these indexing models; thus, the retrieval process considers only the content aspect of scientific documents [2, 3]. However, both structural and content information in these documents enable developing different data management scenarios for which it is sensible to consider structure feature alone, content feature alone, or features of either type. This is mainly challenging because documents may contain shared structures across different topics or common topics, even though fitting into similar structural types [4, 5].

The users of a DL face another challenge in the retrieval process, which is also rooted in the lack of structural organization of text in documents. Traditional IR systems return the whole relevant documents and leave the task of finding relevant information within the documents to the users [6, 7].

Most text documents have structural units, such as headings and sections, commonly represented by explicit markup [8, 9]. Exploiting structural characteristics of structural units or elements can motivate users to study the element content and consider it as relevant [10].
XML as the most popular markup language is originally designed to meet the challenges of large-scale electronic publishing data [11]. This format is perfect because its flexibility handles the profitability of both structured fields and unstructured text components of a content-centric document [12]. Results of [13] show that smaller sections of the articles preferred to be used by researchers as their information source. These preferences can be met using the hierarchical structure of XML documents. Such a structure of a document is strongly valued and gives an overview of logical structure to users to find the desired sections of information [14]. Besides that, the hierarchical structure of XML documents allows the representation of documents with hierarchical levels of granularity to improve precision through focused retrieval, denoting more requirements on the retrieval and representation mechanisms [2]. Retrieving information from XML documents through the IR techniques is called XML element retrieval, proposed in INEX 2005 [7]. An XML element retrieval system retrieves the most relevant XML element of a document based on both CAS features [15, 16]. In other words, users can access a document's contextual elements through a known structure [17].

IR has presented several techniques for the processing and retrieval of relevant documents [18]. As in the last few years, we have observed document clustering as a proliferation solution in applying IR based on the assumption that if a document is pertinent to a query, other documents of that cluster can pertain to that query too [19]. It is believed that data storage indexing can be improved via grouping XML documents together which will have a positive effect on the retrieval process [20]. Also, INEX 2009 proposed a clustering track that follows the hypothesis of clustering in IR application [15]. This hypothesis was used to improve an IR system and satisfy our objectives under the proposed clustering framework.

1.1 Objective

The main objective of this study is to address the mentioned problems and develop an integrated framework to organize a text-rich document collection efficiency through clustering approach, as all information of documents analyzed and their meaningful structural units can be retrieved efficiently based on desired content and structures of users. The retrieved units are meaningful structural units instead of whole documents to help users to find their relevant information more quickly. Indeed, a framework with capabilities of considering structure as well as focused access.

1.2 Contribution

In this study, a scientific XML document collection is produced, containing long text documents of a real DL to meet two challenges of the IR system of a DL simultaneously in one application. This leads to considering structural features in the IR process on the one hand, and enhancing capability of focused retrieval instead of retrieving whole long text documents on the other. Contributions of this study are listed as follows:

1) A new clustering framework was proposed for the next generation of IR systems. This framework puts similar XML elements of documents in clusters that have both similarities in content and structure (CAS).
2) Two mentioned problems of IR systems, I: ignoring structural features and II: retrieving whole long documents instead of a short relevant part have been considered in one application simultaneously.
3) The applied XML document collection of this study is a unique full-text XML document collection, containing documents from different data sources with different structures and content that can evaluate our proposed framework from different aspects.
4) Proposing a new similarity function to measure the similarity between textual element sections.

2.0 RELATED WORKS

Document clustering is an important research topic that is useful in different areas such as IR, web mining, text data analysis, etc. The increasing representation of documents through XML is led to pivotal development in different areas of knowledge discovery, data management, and especially IR. The process of XML document clustering is a process including different approaches each of which is important. This study briefly reviews different approaches in this area.

2.1 XML Document Clustering Based on Content and Structure

In XML clustering approaches, Doucet et al. [21] proposed a method based on vector representation through the K-means technique. In this method, an XML document is transformed into a vector, taking CAS information into account. Two text and label feature sets are generated from XML documents: bag of tags and bag of words that can be handled separately or merged. In this approach, a “textitude" measure was introduced to compute the ratio between content and structural information weight. The proposed approach implies more acceptable results than other proposed clustering methods in INEX 2006. The drawback of this method is ignoring the structural relevancy of the XML documents.

Another significant XML clustering was presented by the authors of [22] is a graph-based approach, which clusters XML documents based on the CAS information through the Self Organizing Map for Structured Data (SOM-SD). Kc et al. [22] extended the SOM-SD method in handling contextual information and proposed the CSOM-SD approach to investigate the effectiveness of contextual information in clustering tasks. Indeed, they applied the CSOM-SD method to process XML information in a contextual manner instead of a casual strict fashion. This approach acquired an international INEX competition prize [23].

Most XML document clustering methods consider both content and structural features in their similarity computation less due to their high processing time [24], especially when we deal with text-centric XML documents. Considering the semantic aspect has a positive effect on the computing similarity between objects of a cluster [25, 26] . However, some studies proposed remarkable methods in clustering based on CAS while considering the semantic aspect of XML documents in clustering [5, 27, 28]. In the proposed approach by Tran et al. [28], both CAS information is represented by the VSM. The structure similarity is computed by co-occurrence and commonality of paths between structures of documents, and content similarity is computed by using the Latent Semantic Kernel (LSK) to specify the associated semantic within the document content. To measure document similarity, the values of structural similarity and content similarity are combined. The pair-wise similarity matrix is the output of document matching, consisting of the document similarity value between each pair of XML documents. Two other tree-based XML clustering methods were proposed by {Tagarelli, 2006 #28;Tagarelli, 2010 #29}[5, 27]. However, these two frameworks are considerable because the problem of semantic clustering is solved, but they introduced the new notion of tree tuple to represent XML elements as transactional data. In this method, each XML document is decomposed into a set of smaller documents, namely tree-tuple, and represents them as transactional data. These produced tree tuples have a semantic correlation underlying the semantics of the original XML document. Each item (tree tuple) of the produced XML transactional domain embeds a combination of CAS features. The XML tree tuples modeled as transactions are efficiently clustered through a clustering algorithm. Evaluation of this framework on real data sets of various domains shows its significance as a solution to cluster XML documents based on the CAS.

The XML document clustering, considering both CAS aspects, is a practical topic in a heterogeneous environment, while structural and CAS methods are not proper in a homogeneous corpus [29]. Some of the novel proposed approaches in this area are reviewed. As a significant one, Bessine et al. [30] proposed XCLSC as a novel XML document clustering approach based on CAS similarity by using tree structured-content summaries. In this approach, first, an XML document is represented as a rooted ordered labeled tree. Then, for reducing the size of the XML tree and extracting structural summaries, nested-repeated nodes and repeated nodes are eliminated. Each non-leaf node that has the same label as its ancestor is a nested-repeated node, and a node whose path has been traversed before is a repeated node. In order to combine both CAS features, level structure-content representation was proposed which, distinct XML nodes at each level of the document, grouped and organized as a vector of levels, as each level consists of some different XML nodes. Each node is represented by its name, its parent, and its textual content, where textual content contains a bag of words without its frequencies. Regarding the tree structural summary and its corresponding content, the dimensionality of the entered document is reduced, which leads to the reduction of processes. Thus, it is appropriate for big datasets. After that, Rezk et al. [31] proposed another considerable clustering XML document based on CAS features which improved the quality of clusters by considering more features from source schemas. The authors of [31] used the XEdge algorithm [32] for structural similarity, and in order to get content similarity, aggregate three similarity measures; Cosine, Jaccard, and Jensen-Shannon divergence. The proposed approach was compared with another XML document clustering approach with the same structural algorithm, while its content similarity was computed by the Jaccard measure. The gained results were similar to [28] and proved the efficiency of the aggregation content similarity measure in a heterogeneous environment, while in a homogeneous XML dataset, the CO clustering approach yields better results than the CAS approach. Also, the results compared with both XEdge [32] and XCLSC [30] and demonstrated better quality clusters.

As another XML clustering work based on CAS, Magdaleno et al. [9] proposed a new similarity function, namely OverallSimSux. The author believed that an XML document comprises Structural Units, each of them called SU. The main XML document collection D is divided into n collections, as n is the same number as SUs of a document. In order to obtain the correspondence between SUs and collection, the K-collection of collection D is introduced. It is formed by a set of Structural units of main documents and represented by DSU_k where SU_K is kth SU of d and d is a document of D. The proposed function analyzed the relationship among the documents and simultaneously treated documents and SUs like indivisible units and independent collections respectively. Similarity matrix of SUs constructed through TF-IDF measure. In order to add document structure in analyzing process, popular VSM modified, and frequency of SUs corresponding analyzed term used for weighting. For producing a similarity matrix, the similarity between two pairs of a document is calculated using a cosine measure. In this approach, for each k-collection, a cluster is produced. Finally, the OverallSimSux matrix is calculated through clustering results of all k-collection and similarity matrices calculated by cosine measure.

There are XML document clustering approaches all focused on the "structure-constrained phrase" method in their process [33-37]. In the first one, the author of [33] used the Non-negative Matrix Factorization (XC-NMF) technique to partition an XML document collection into topically homogeneous groups by structure-constrained n-grams. By this means, Subsets of XML documents that are related semantically can be separated. NMF is performed by using the alternating least square method that combines expedients to decrease the amounts of factorization, especially in large-scale collection. In the Second one, Costa et al. [34] proposed the XCO-CLUST approach (XML CO-Clustering) method and claimed that XML CO-clustering is more effective in partitioning XML documents than XML Clustering. CO-Clustering is a two-side clustering approach [38] which is a powerful refinement of the original clustering task. In this approach, the interrelations between XML documents and their own features are exploited while simultaneously both of them are clustered in an interactive environment. The main technique of this approach is based on non-negative matrix tri-factorization. This study considered the relevancy of both CAS features of XML documents in their experiments and achieved higher effectiveness in comparison with XML clustering. Both XC-NMF and XCO-CLUST approaches are particularly useful when we deal with substantial text-centric XML document collections. The third significant approach is XPart (XPartitioning), an XML document clustering approach based on partitioning proposed by [35]. In this study, bag-of-words representation is considered a problem that is not reliable for discovering topical relatedness of XML documents because a particular sequence combination of words can have a different meaning in the context of each substructure. The XPart approach was developed based on structure-constrained phrases, inspired by bag-of-phrase to preserve the text's meaning more. First, Structure-constrained phrases are estimated by word n-grams in substructures from root to leaf paths with fixed length. The word n-grams are considered as XML features. Then, the main XML collection is summarized over gained XML features and represented as a collection of transactions. In order to meet the dimensionality, a new weighting scheme is used as a feature selection technique to select XML features based on their relevance to clustering. In the last step, an appropriate partitioning algorithm is applied for partitioning the collection of XML transactions over fixed length word n-grams. The XPart approach is extended later in [36], and XML transactions can be partitioned over both fixed and mixed length word n-grams. This is achieved by introducing a criterion to more prune the selected XML features automatically and maintain those that represent a suitable trade-off of relevancy between clustering and phrase-level meaning extent, obtained using the respective contextualized word sequences length. The experimental results on real-world XML corpus dedicated higher effectiveness in partitioning the clustering process than the conventional approach. In the study of [37], authors performed to investigate the structure-constrained phrases benefits in the processing of XML document clustering from the perspective of described approaches XC-NMF, XCO-ClUST, and XPart. Triple approaches packaged together and formed a cohesive study that improves prior ones. As one example of these improvements, XC-NMF and XCO-CLUST methods are generalized by structure-constrained n-grams to better understand their capabilities. In addition, the new class of XML features produced in the XPart approach applied in XC-NMF and XCO-CLUST approaches led to achieving better results. Furthermore, an appropriate feature selection technique is presented to meet the challenge of dimensionality by filtering some XML features based on their clustering relevance. The experimental evaluation showed that the results improved more than those in the original studies.

Another CAS clustering method proposed by Al-Shammari et al. [24] is considerable in process speed named XclusterMaint. The proposed framework decreases the computational cost in a dynamic environment through the maintenance of existing clusters when new documents arrive. The applied data set of this work was a data-centric XML document collection, and content similarity is limited to only structured data.

In recent years, Dongo et al. [29] proposed a remarkable framework that enriches the content and structural similarity measure using semantic analysis. The framework was based on the optimization of the Latent Semantic Indexing [33] named *LSI, in which term occurrence is weighted regarding the context of its document in the term-matrix generation phase. Considering semantic analysis led to better document classification.

2.2 Xml Document Clustering Based on Only Structural Features

The researches of [39, 40] are two other studies in the XML document clustering area that concentrated on clustering by pattern instead of relying on pairwise similarity measure. In the study [39], the authors proposed to utilize maximal frequent subtrees for clustering XML documents by Satisfy/Violate algorithm. In this approach, first, the whole XML collection is mined for patterns with maximal frequent subtrees. Then, the patterns are clustered into k user-defined clusters through the AHC algorithm. In order to produce a similarity matrix for each pair of patterns, the number of common documents is used as the similarity measure between patterns. Finally, each document is assigned to the first cluster to satisfy it using the satisfies/violates operator. More generalization of this approach led to another clustering framework based on a pattern called XPattern, proposed by [40]. The proposed framework is a four-step framework, including choosing a pattern definition, pattern mining, pattern clustering, and document assignment. In this framework, tasks of pattern clustering and document assignment are combined, leading to grouping documents based on their features instead of their direct similarity. This framework is evaluated using PathXP, an algorithm based on the maximal frequent path. The experimental results on different pattern definitions demonstrated that frequent paths produced highly qualified clusters with reasonable efficiency. Both researches are distinguished studies in XML document clustering based on only structure.

2.3 XML Document Clustering Based on Only Content Features

There are other studies like [41] in web mining with a concentration of XML document clustering based on content to find the more interesting XML documents on the web. Corpus like Wikipedia, which exist on the web, are huge and classified as big data collection. Finding similar features for such a collection takes lots of time. In order to solve this problem, Muralidhar et al. [41] proposed a hybrid approach that first finds frequent XML documents through the Apriori algorithm of Association rule mining and then clusters XML documents by the popular K-means clustering algorithm. This approach can be useful for clustering of XML documents in the web environment and improving retrieving information in the web, but ignoring infrequent documents is not appropriate for an static XML document collection like DL because each document in the collection may meet the need of a user. This study focused on only the content feature of an XML document.

2.4 Improving Document Clustering Process

Text document clustering faces different problems such as high dimensionality, time-consuming, uninformative, and sparse features that have negatively affected efficiency, time complexity, accuracy, etc. Some researchers dealt with these problems to improve document clustering methods. [42, 43] are two studies that address the problem of sparsity and dimension reduction respectively through transforming document data to structured Document-Term Matrix (DTM) data which is a matrix based on term-frequency. Since the converted DTM has columns larger than rows, it is sparseness. Jun et al. [43] proposed a new hybrid SVD-PCA method to address the sparseness of data in clustering of document data. PCA (Principal-Component Analysis) is an efficient dimension reduction method for K-means clustering [44] and SVD (Singular Value Decomposition) is an efficient method for document classification [45]. In order to produce effective clusters, support vector clusters and Silhouette measure are applied to the popular K-means clustering method. Bafna et.al

[42] constructed a phrase document matrix based on a synset grouping of semantically similar phrases so that, reduce dimensionality and as a result more accurately clusters.

Two remarkable studies in improving the clustering process are [46, 47] with a concentration on the time factor. Ding et al. [46] proposed a new method for clustering XML documents based on frequent patterns. In this study, first, encode the XML documents by a coding tree structure and translate frequent pattern mining from XML documents into frequent pattern mining from the string. Then, a cosine similarity method and a hierarchical clustering algorithm are used to cluster a collection of XML documents by frequent patterns. The advantage of this study is reducing time consumption of calculation differences between each pair of XML document collections since frequent patterns are proper subsets of the main collection and the time of similarity calculation decreases. The proposed method of this study is appropriate in the phase of similarity computation of an XML document clustering project. Lydia et al. [47] proposed a clustering method for large-scale documents, implemented by the Hadoop platform through the Map-reduce technique. In this study, the matrix format of documents produced through LSD and SVD and then the NMF method, optimized with new rules is exploited in the process of feature extraction matrix. The performance of the proposed method is increased through reduction of computational time. Applying big data solutions and optimized NMF methods are the strength points of this study.

In the study of [48], authors accomplished a comprehensive survey and experimented to prove the proficiency of metaheuristic algorithms, known as nature-inspired optimization algorithms, in improving text-based document clustering algorithms. To that end, [49] applied a Fruit-Fly Optimization (FFO) as a metaheuristic algorithm and combined it with the K-means algorithm to face up to the problem of local optima in text document clustering. These clustering approaches are good ideas for unstructured documents and can be utilized in only analyzing the unstructured part of a semi-structured document collection.

3.0 MATERIALS AND RESEARCH DESIGN

In this research, a new framework has been proposed based on the clustering to cluster XML elements of documents of a collection according to both CAS similarities. Fig. 1 depicts the outline of the proposed framework.

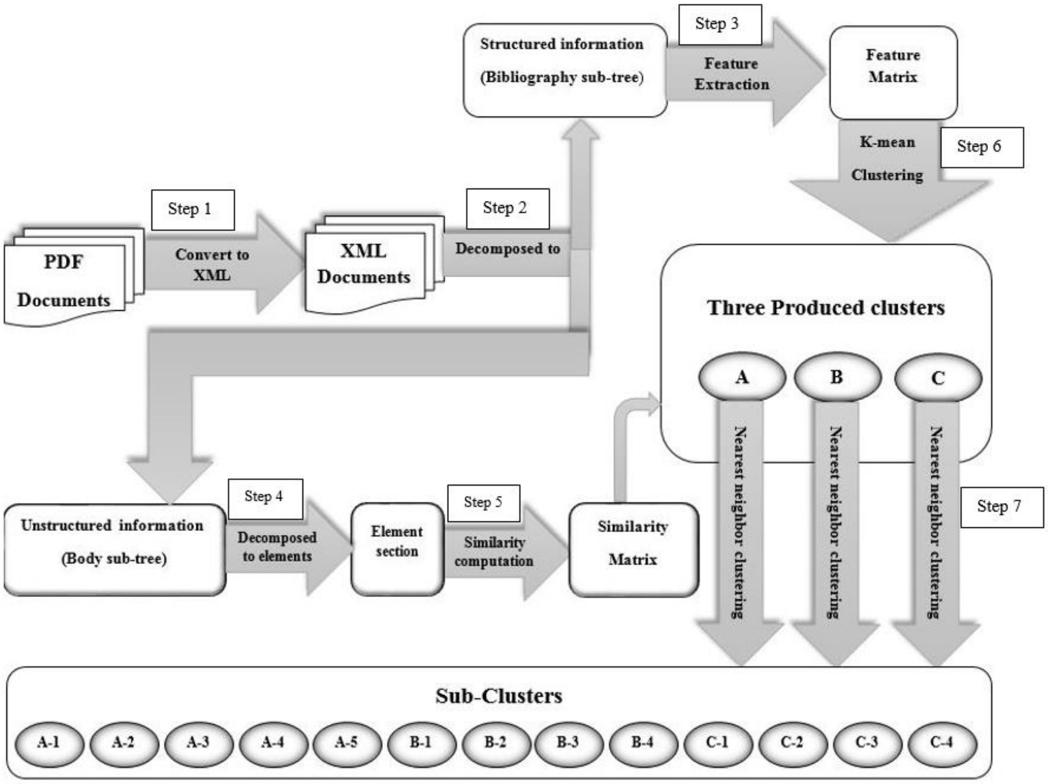

Fig. 1: Outline of the proposed framework

The following are seven steps of the proposed framework:

1) In the first step, document collection is converted to semi-structured XML format for the structural organization of their text.
2) For text documents that contain a mixture of structured and unstructured content, these two types of information are decomposed to process separately.
3) The features of structured information of each XML document are extracted to produce a feature matrix. This feature matrix is used as input for the K-means clustering algorithm.
4) The unstructured information part of each document is decomposed into smaller granularities using the hierarchical structure. The decomposed elements allow users to attain a focused access strategy.
5) The similarity between the extracted elements is computed by the TF-IDF and CosSim functions.
6) The XML documents are clustered based on their structure using the feature matrix and K-means clustering algorithm.
7) Finally, constitutive element sections of documents for each primary cluster are clustered again using the similarity matrix. The second level of clustering is based on the content using the nearest-neighbor clustering algorithm. Final clusters contained elements that are similar in two aspects of CAS.

3.1 Data Set

To test the proposed framework, a full-text XML document collection in the scope of a Digital Library (DL) was needed. The XML document collections PubMed, DBLP, IEEE version 2.2, and INEX IEEE are in the domain of DL. Each of these collections has some characteristics of a variety of data sources, a variety of structures, and a variety of content and focused access information. All of these properties are needed to reach the goal of this study. The characteristics of XML document collections in Table 1 were reviewed briefly. In Table 1, a variety of data sources means documents from different publishers. Different data sources published their documents in their own schema. Therefore, it leads to differences in structures. Variety of structures means that the collection should contain documents from different types like journals, conferences, etc. Variety of data sources and variety of structures lead to diversity in structure from all aspects. Variety of content means documents are related to different subjects, which helps to evaluate IR systems from both CAS information. "Focused Access" means which information part of a document is accessible. They may contain only bibliography information, abstract, or full text of them. This property is needed for focused retrieval capability.

Table 1: Characteristics of available XML documents

Property collection	Variety of structure	Variety of content	Variety of data sources	Focused access
DBLP	✓	✓	✓	×
PubMed	✓	✓	✓	×
IEEE version 2.2	✓	✓	×	×
IEEE INEX	✓	✓	×	✓
Produced XML collection	✓	✓	✓	✓

Each of these XML collection is well structured. However, regarding Table 1, there are deficiencies in focused access properties in DBLP, PubMed and IEEE version2.2 and a variety of data sources in IEEE INEX. Therefore, none of them satisfy our research needs in the domain of a DL. The produced XML collection has the characteristics of a real DL. It was composed of full-text scientific articles from different data sources and structures from various content areas. Thus, it can prove the efficiency of the research method in clustering based on the CAS. Such a full-text XML document collection was created by Samadi [50] through converting a PDF document collection to XML collection. This data set comprises three different structures from two data sources; IEEE and Elsevier, totaling 1690 full-text XML documents in 90 MB size which is available from 2013. Such selection was implemented so that our collection is comprehensive and has the characteristics of a real DL. More detailed characteristics of this data set are presented in Table 2. The produced XML collection has some weaknesses. It is small and can comprise more data sources and more article types.

Table 2: Number of XML documents of the data set in this research

	Structure 1	Structure 2	Structure 3
Data Source	IEEE		Elsevier
Article Type	Journal Article	Conference & Proceedings	Journal Article
Numbers	641	402	647

The structure of an XML document is defined through DTD. A well-formed XML document collection facilitates its analyzing in structural similarity computation while this task is complicated in XML documents with different DTDs. The applied XML collection in this study has a single DTD.

3.2 XML Tree-Structure Generation (Step 2 Of Fig. 1)

Text documents often contain a mixture of structured and unstructured information [51]. These two types of information have been organized in our XML documents. To capture both CAS features of these two types of information, their information should be parsed first and represented using a data model such as the tree structure. By using the DOM parser, each XML document was represented in a Data Tree (DT) model. A DT is a labeled ordered tree constructed from one root and nodes that are labeled. We represented each XML document in a tree structure in Fig. 2.

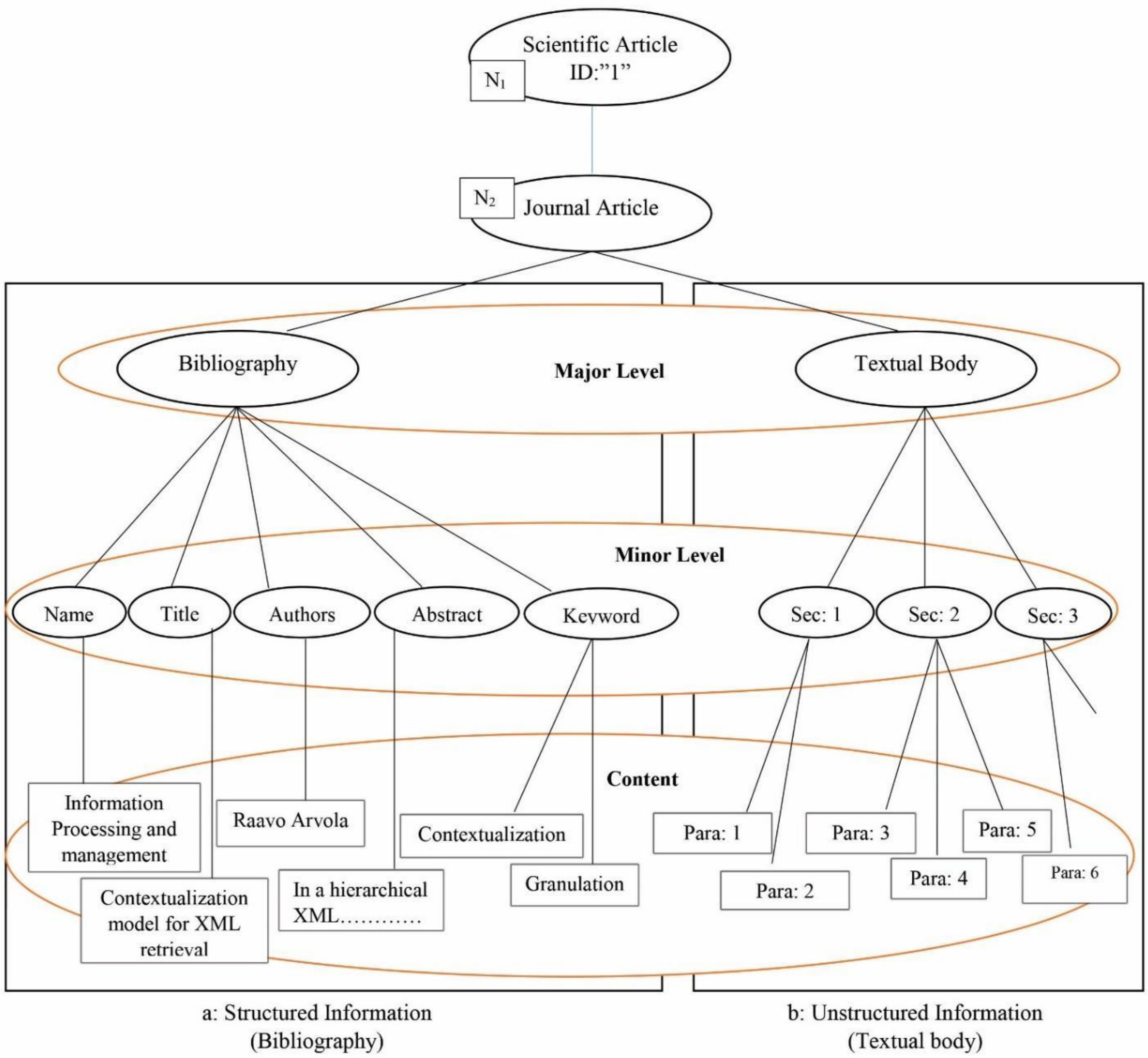

Fig. 2: A Sample of DT of an XML document tree corresponding to data representation phase

As shown in Fig. 2, the represented XML tree document has a root (node N1) with a property named Article ID. The value of this property is unique for each XML Tree Document (XTD). This value is the identifier of sub-trees and XML elements. The second node of this DT is N_2, the value of which is different for each type of scientific article. The string value of the N_2 node is important in identifying different structures, such as journal articles, conference papers, and proceedings. Following node N2, the XTD is decomposed into two main sub-trees, left and right, based on two types of information. These two types of information are recognizable in Fig. 2(a) as structured information (bibliography) and Fig. 2(b) as unstructured information (body). Three phases of clustering, including data representation, similarity computation, and clustering, were performed separately for these two types of information.

In the following, in phase 1, Data Representation, representation and extraction of structured features and unstructured features will be shown in sections 3.3, 3.4 and 3.5 respectively. Then in phase 2, similarity computation, first, the similarity of the structured information features will be computed in section 3.6 and then, the similarity of element sections of the unstructured information part will be computed. Finally, in phase 3, two levels of the proposed clustering solution will be implemented in the 3.8 and 3.8 sections.

Phase 1: Data Representation

3.3 Representing Structured Information (Bibliography).

As shown in Fig. 2(a), the information of the bibliography was located in two nodes; element nodes, corresponding to tag names of the XML documents, and textual nodes, corresponding to the Textual Content Units (TCU) existing in the terminal leaf nodes. In order to extract features of bibliography information of all XTDs, a jungle consisting of bibliography of all XTDs with the same root "Scientific Article" was constructed. Fig. 3 shows a part of this jungle as a sample, consisting of two different kinds of XML documents as an example.

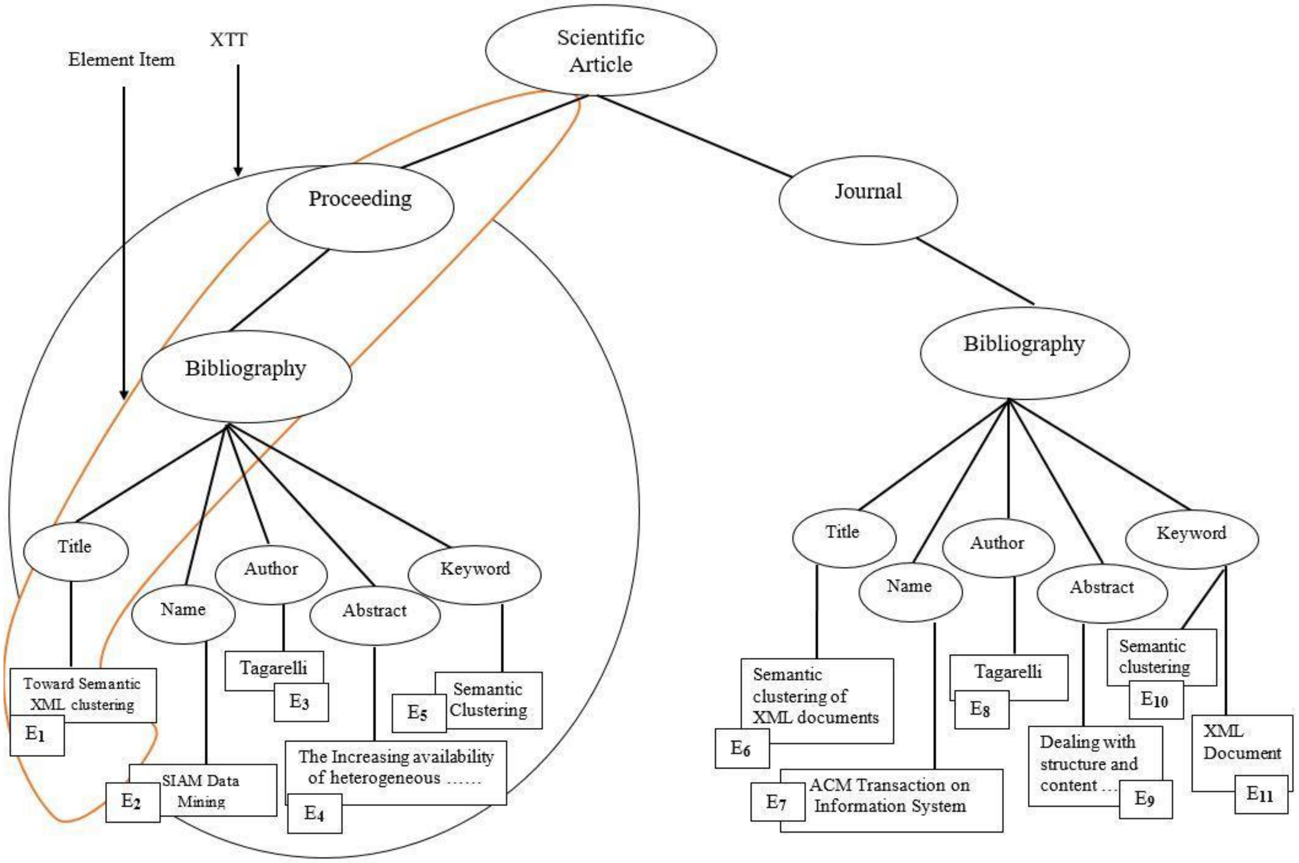

Fig. 3: Schema of the jungle comprises XTT and the XTTs comprise element items.

With respect to the notion of the tuple in transactional data, the name XML Tree Tuple (XTT) was assigned for each bibliography in Fig. 3. The notion of XTT and its processing as a transaction was proposed by [27]. Features of each XTT are determined by the XML paths and their respective answers. An XML path comprises a sequence of tag names, attributes, and strings. The first node in each XML path is equivalent to the root node of the jungle. Each XML path yields an answer (AXTT (P)). The answer of each XML path is a Textual Content Unit (TCU) or #PCDATA corresponding to the terminal leaf node of that XML path. Structure features were extracted from the XML paths, and content features were extracted from the TCUs. Each XML Path with its respective answer is named an element item identified in Fig. 3, as [E_1, E_2…E_{11}]. These element items are uniquely indexed. Indeed, each E in an XTT consists of two interdependent attributes: XML path and its answer. To clarify further, all extracted XTTs and element items of Fig. 3 are shown in Table 3 as transactional data. The columns indicate three attributes XML Path (PXTT), answer of XML Path (AXTT (P)), and element item (Item ID). As such, all XTTs are mapped to transactional data. For example, in Fig. 3, the left XTT consists of five element items, namely E_1 to E_5. The first element of this XTT consists of a path [Scientific Article. Proceed. Bibliography. Title] and answer of the path "Toward semantic clustering." Subsequently, content and structure features were extracted from these transactional data.

Table 3. Features of each XML Tree Tuple (XTD)

(XML Tree Document) XTD # ≡ (XML Tree Tuple) XTT#: 1		
PXTT=1(P) ≡ Structure (S)	AXTT=1(P) ≡ Content (C)	Leaf Node ID/Item ID
S_1 = Scientific Article. Proceed. Bibliography. Title	C_1 = Toward Semantic XML Clustering	E_1
S_2 = Scientific Article. Proceed. Bibliography. Name	C_2 = SIAM Data Mining	E_2
S_3 = Scientific Article. Proceed. Bibliography. Authors	C_3 = Tagarelli	E_3
S_4 = Scientific Article. Proceed. Bibliography. Abstract	C_4 = The increasing availability of …………………………	E_4
S_5 = Scientific Article. Proceed. Bibliography. Keyword	C_5 = Semantic Clustering	E_5

(XML Tree Document) XTD # ≡ (XML Tree Tuple) XTT#: 2		
PXTT=2(P) ≡ Structure (S)	AXTT=2(P) ≡ Content (C)	Leaf Node ID/Item ID
S_6 = Scientific Article. Journal Article. Bibliography. Title	C_6 = Semantic Clustering of XML document	E_6
S_7 = Scientific Article. Journal Article. Bibliography. Name	C_7 = ACM Transactions on Information System	E_7
S_8 = Scientific Article. Journal Article. Bibliography. Authors	C_3 = Tagarelli	E_8
S_9 = Scientific Article. Journal Article. Bibliography. Abstract	C_8 = Dealing with structure and content …………….	E_9
S_{10} = Scientific Article. Journal Article. Bibliography. Keyword	C_5 = Semantic Clustering	E_{10}
S_{10} = Scientific Article .Journal Article. Bibliography. keyword	C_9 = XML document	E11

3.4 Feature Extraction Of Structured Information (Bibliography)

XML paths characterize the ordinary bases for extracting structure features. Each extracted XML path is indexed to S_k, where K is a unique value. In the process of indexing structure features, the extracted XML path is first compared to the previously extracted structure. If it is identical to one of them, the same index is allocated to both of them. This indexing leads to a decrease in storage space. For example, the structural feature of E_{11} from Table 3 has a similar structure to the structure feature of item E_{10}. In such conditions, S_{10} is allocated to both structural features instead of allocating a new index to E_{11}. The exact process is performed for indexing the content features. The extracted contents of each item (#

PCDATA) are indexed to C_k, in which K is unique. As such, all the XML tuples of all the XML documents are traversed, and their content and structure features are extracted to produce a feature matrix similar to Table 4.

Table 4. Produced indexed feature matrix from the extracted structure and content features.

	PXTT=1(P) ≡ Structure (S)										AXTT=1(P) ≡ Content (C)								
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
XTT# : 1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0
XTT# : 2	0	0	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	1	1

The columns of the produced matrix in Table 4 consist of the indexed content and structure features, while the rows are related to each XTT. If an XTT has a feature, the related column is filled up with the value "1"; otherwise, its value is "0". This feature matrix was used in the first level of our K-means clustering framework to cluster the XML documents based on their structure.

The extracted features are so huge that exploitation of such a huge amount of data needs reliable methods of feature selection to support the extraction of useful features from the original ones by removing impertinent features[52]. In order to get better quality clusters, a feature selection task was carried out before clustering. In this task, features that were meaningful owing to their low frequency in the whole collection were removed from the feature matrix. As such, the structured information in an XML document is represented, and its features are extracted.

3.5 Representing Of Unstructured Information (Textual Body)

Unstructured information of a document is its flat textual body part. This information is represented in the right subtree of Fig. 2(b). This information is usually time-consuming for users to find their desired information part. Thus, we used the hierarchical structure of an XML document and decomposed this long block text to smaller granularities to achieve better precision by means of focused retrieval mechanism. As shown in Fig. 2(b), the XML document was structured in three hierarchical levels major, minor, and content levels, which led to accessing granularities of the whole textual body, section, and paragraph. Levels of granularities divide a text document into elements that cover all the text content and retrievable elements are not structurally overlapping. Retrievable units are available by hierarchically decomposing a document into sections and paragraphs as smallest Textual Content Units (TCUs). Granularities have hierarchical relations to each other. These hierarchical relations help to access them.

Each granularity is built based on the TCUs. Thus, the textual content of each paragraph (as the smallest unit for representation of the unstructured information part of a document) should be preprocessed and cleaned. This text preprocessing task is greatly important in making the clustering process easier and attaining clusters of better quality. Text preprocessing is performed by lexical analysis, which mostly includes three tasks i.e. tokenization, stop word removal, and stemming [53]. Using the stop word removal, many unessential terms are removed and by stemming, the words are reduced to their root form or their stems. The stemming was implemented based on the most popular stemming algorithm proposed by Porter [54]. The lexical analysis led to dimension reduction of text, text cleaning, and preparing for the similarity computation phase.

Phase 2: Similarity Computation

3.6 Similarity between the Features of Structured Information (Feature Matrix).

Clusters are produced based on the number of their common features so that the similarity of objects inside one cluster increases whereas the similarity between the objects of two distinct groups minimizes [55, 56]. The similarity between the XTTs is computed through the clustering algorithm. The bases of computing similarity are the minimum square distance between each new observation and the mean of each cluster. This similarity was computed by using the K-means clustering algorithm.

3.7 Similarity between Element Sections of Unstructured Information Part (Body Part).

Most users are interested in looking for section elements of journal articles [57]. Hence, we choose sections as the desired granularity of users for clustering and retrieving information. Each element section was captured easily from the collection of its preprocessed paragraph, and the similarity between each pair of them was computed.

In order to compute the similarity between the textual element sections, a syntactic function is devised to measure the relevancy of terms in the element sections. The well-known weighting function TF-IDF (Term Frequency-Inverse Document Frequency) [58] is the best token-based similarity measure which is typically used in text mining and IR approaches [59]. TF-IDF includes two statistical criteria: term density in the given text and term rarity in the text collection. The underlying idea is that those terms repeated more often in a document are considered more important, i.e. more indicative of the topic, and terms that appear in many different documents are less indicative of the overall topic [60].

Representing the element section of documents, instead of the complete documents, led to the consideration of terms in the element sections instead of terms of documents in the TF-IDF weighting function. Thus, the domain of this weighting function was changed and a new TF-IEF (Term Frequency-Inverse Element Frequency) weighting function was defined with the new definition of its two criteria; (1) term density is defined as total frequencies of a term occurring in an element section, and (2) term rarity is defined as the total number of a term repeated in the entire element section collection. Using the TF-IEF weighting function, the term relevancy increases with the frequency of the term within the local textual element sections, with the term rarity across the entire collection of textual element sections. The weighting function of TF-IEF for each extracted term within each element section is computed by Equation 1.

$$W_{ij} = TF_{ij} * IEF_i \qquad \text{Equation 1}$$

where IEF_i is the inverse element section frequency of term i that is obtained by $IEF_i = \frac{N}{EF_i}$ where EF_i is the element section frequency of term i, which is equal to the number of element sections containing term i, and N is the total number of sections in the entire element section collection. Finally, a logarithm takes the quotient in order to moderate the effect relative to TF. By this means, all extracted terms are weighted by TF-IEF in Equation 1 to compute the similarity between each pair of element sections.

Similarity between the element sections is computed through a similarity function. The similarity function computes the degree of similarity between two vectors. There are some popular similarity functions, such as the Inner Product and the Cosine and Jaccard Coefficients [61, 62]. The Cosine similarity function is most commonly used in high-dimensional spaces, which is suitable for text IR [63]. In this study, the Cosine similarity measure was computed between each pair of TF-IEF weighted vectors of element sections j and k through Equation 2.

$$CosSim(E_j, E_k) \; = \frac{\sum_{i=1}^{t} \; (w_{ij}. w_{ik})}{\sqrt{\sum_{i=1}^{t} \; w_{ij}^2 . \sum_{i=1}^{t} \; w_{ik}^2}} \qquad \text{Equation 2}$$

where E_j and E_k are two pairs of element sections, W_{ij} is the TF-IEF weighted vectors of element sections j, and W_{ik} is the TF-IEF weighted vectors of element section k. Through the cosine similarity function, the similarity between each pair of element sections was computed, and a similarity matrix was produced. The range of values of a similarity matrix is between [0, 1]. The CosSim value is equal to one for two of the same element sections and less than one for any other pair of element sections. The lesser angle of similar element sections leads to a higher $CosSim$ value. Thus, more similar element sections have higher $CosSim$ values in the produced matrix. This similarity matrix is the input of the next section of clustering.

Phase 3: Proposed Clustering Solution (CAS)

The clustering of XML data groups similar XML documents/elements together within a single cluster, while placing dissimilar XML documents/elements in different clusters [2]. In order to cluster XML elements of documents based on both their CAS features, a clustering solution is proposed in two levels. In the first level, all XML documents are clustered structurally using their structured information and K-means clustering algorithm. In the second level, XML elements of each produced cluster are clustered separately again based on the content similarity using the nearest neighbor clustering algorithm.

3.8 First Level oF Clustering Using K-MEANS.

In the first level, a k-means clustering technique is applied in order to cluster XML documents using the produced feature matrix. In the K-means clustering algorithm, a parameter K should be defined in order to partition the collection into K clusters. The values of this parameter are very important in the result of clustering; thus, they should be defined accurately. We determined K based on the variety of structural types to evaluate our results based on the structural features. Our document collection contains three types of conference, journal, and proceeding structures; thus, the value of K=3 was allocated as the number of clusters in the K-means algorithm. The process of K-means is begun by selecting K observation randomly as each of them initially represents a mean as the centroid of C_j. Other remaining observations are assigned to the clusters on the basis of the highest similarity to the centroid of a cluster. The highest similarity is computed with respect to the minimum square distance between an observation and cluster mean computed by Equation 3.

$$Arg_{\ C}\ Min \sum_{j=1}^{k} \sum_{x_{i \in C_j}} \left\|x_i - m_j\right\|^2 \qquad \text{Equation 3}$$

where K is the number of clusters, Xi is indicated as ith observation, and C_j is represented as the jth cluster. The m_j is the mean vector of the jth cluster. The mean vector of a cluster is the centroid of a cluster, which is changed in each iteration. After assigning the new observation to the clusters, a new mean is computed as a centroid for clusters. This process iterates until the centroid of the clusters is fixed. By applying this clustering algorithm, K clusters are produced, similar XML documents are located in the same cluster and all the XML documents are clustered structurally. The result of this primary clustering is k clusters. Each considered a separate collection for the second level of the clustering solution.

3.9 Second Level of Clustering Using The Nearest Neighbor.

In this section, each K-produced cluster is considered as a separate collection. Each of these collections is clustered again separately in the second level. In the second level, all the concatenated element sections of XML documents are considered as new observations. Thus, XML elements that were clustered structurally in the first level are clustered again in the second level based on their content using the produced similarity matrix and a hierarchical clustering algorithm. The nearest neighbor clustering algorithm is used in this process because it is suitable for textual data [70]. The nearest neighbor is a hierarchical system of clustering, which successively merges textual elements into clusters based on pair-wise similarity [2]. The main idea of the nearest neighbor is to find similar pairs of clusters to be merged. The nearest neighbor clustering successively follows a chain of clusters A → B → C → …, where each cluster is the nearest neighbor of the prior one, up to the level that a pair of clusters is reached that are mutual nearest neighbors. This algorithm is performed using the values of the Cosine similarity matrix.

4.0 RESULTS AND DISCUSSION

Clustering results were evaluated using an external criterion. Regarding the fact that the characteristics of none of the existing XML document collections are compatible with the produced XML document collection applied in this study, and therefore, comparing its results with the existing ones is not reasonable. In order to evaluate the efficiency of structural features, the proposed CAS clustering approach was compared with the CO clustering in a similar situation.

4.1 Evaluation Criteria for Clustering

In this study, the quality of XML clusters was evaluated by an external criterion. In this approach, a set of classes are used as an evaluation benchmark called the golden standard classes. The golden standard classes are ideally produced by human judges with a good level of inter-judge agreement. The external quality is computed to evaluate how well the clustering matches the golden standard classes. The XML document is first classified based on the structural features of a human judgment. We used this classification to evaluate the quality of clusters in the first level. Then, the produced classes were reclassified based on their content relevance. The final classes based on both content and structure were applied in the evaluation of the final produced clusters.

There are external quality measures, such as Purity, F-Score, and Entropy [2]. We used the Entropy metric to evaluate clusters because this metric is widely used in the evaluation of clustering approaches [2]. The Entropy metric is defined for measuring the quality and accuracy of clusters. It measures how different classes of XML documents are distributed in each cluster. The Entropy of produced clusters is computed by Equation 4.

$$Entropy\ (C_i) = -\frac{1}{log\ log\ q}\sum_{r=1}^{q}\ log\ log\ \frac{N_i^r}{N_i} \qquad \text{Equation 4}$$

Where

C$_i$ = ith produced cluster

q = number of gold standard classes

N_i^r = Number of XML data of rth class that is assigned to the ith cluster

N$_i$ = size of the ith cluster

The total Entropy of clustering is defined to aggregate the sum of Entropy weights of each individual cluster according to the size of each cluster. The total Entropy of clustering is computed by Equation 5.

$$Total\ Entropy = \sum_{i=1}^{k}\ \frac{N_i}{N}\ E\ (C_i) \qquad \text{Equation 5}$$

Where

K = the number of produced clusters

$$N_i = size\ of\ i^{th}\ cluster$$

N = sum of the size of each produced cluster or size of XML objects in the whole collection

A perfect clustering approach produces clusters that include documents from a single class and the Entropy result will be zero. Smaller Entropy result shows the better quality of clusters.

To clarify how to calculate the Entropy of produced clusters, all steps of calculation based on the example schema of Fig. 4 and Fig. 5, represent golden standard classes and produced clusters respectively demonstrated as follows.

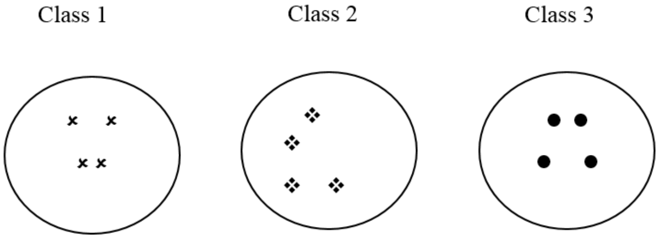

Fig. 4: Golden standard classes

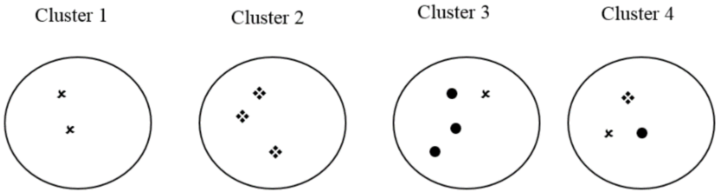

Fig. 5: Produced clusters

At first, Entropy is computed for each produced cluster using Equation 4, as follows.

$$Entropy\ (C_1) = -\frac{1}{log\ log\ 3}\sum_{r=1}^{3}\ log\ log\ \frac{N_1^r}{N_1} = -\frac{1}{log\ log\ 3}\left(log\ log\ \frac{2}{2}\right) = 0$$

$$Entropy\ (C_2) = -\frac{1}{log\ log\ 3}\sum_{r=1}^{3}\ log\ log\ \frac{N_2^r}{N_2} = -\frac{1}{log\ log\ 3}\left(log\ log\ \frac{3}{3}\right) = 0$$

$$Entropy\ (C_3) = -\frac{1}{log\ log\ 3}\sum_{r=1}^{3}\ log\ log\ \frac{N_3^r}{N_3} = -\frac{1}{log\ log\ 3}\left(log\ log\ \frac{1}{4} + log\ log\ \frac{3}{4}\right) = 1.523$$

$$Entropy\ (C_4) = -\frac{1}{log\ log\ 3}\sum_{r=1}^{3}\ log\ log\ \frac{N_4^r}{N_4} = -\frac{1}{log\ log\ 3}\left(log\ log\ \frac{1}{3} + log\ log\ \frac{1}{3} + log\ log\ \frac{1}{3}\right) = 3$$

And then total Entropy of all clusters is computed using Equation 5, as follows.

$$Total\ Entropy = \sum_{i=1}^{4}\ \frac{N_i}{N}\ E\ (C_i) = \left(\frac{2}{12} * 0\right) + \left(\frac{3}{12} * 0\right) + \left(\frac{4}{12} * 1.523\right) + \left(\frac{3}{12} * 3\right) = 1.257$$

4.2 Evaluation Of K-Means Clustering Corresponding To Structure Only Clustering

The quality of data mining technique (e.g. clustering) is very reliant on the noisiness of the used features for the clustering process. Thus, the features should be selected efficiently [64]. Clusters are produced based on the number of their common features. The low frequency of a feature means the feature is not common in any other XML file or is common in a few XML files. Thus, that feature cannot be effective in computing the similarity between XML files. In order to yield better similarity computation and better result, features that were meaningful owing to their low frequency in the whole collection were removed so that appropriate features are selected. Then, K clusters were produced using the remaining effective features. The Entropy results of the three produced clusters and their size and accuracy are presented in Table 5.

Table 5: Characteristics of the produced K-means clusters

	Cluster A	Cluster B	Cluster C	Total Clustering
N = Cluster size	119	40	128	287
Entropy	0	0.0230	0.0143	0.0096
Accuracy	1	0.977	0.9857	0.9904

Those Entropy result values of clusters that are zero or very close to zero show that constitutive observations of clusters were distributed properly as their accuracy is more than 97% (in total, close to 99%). Getting such proper results indicates the success of our method in selecting effective features and mining structured information. Another reason to get such results is the production of XML document collections with identical DTD.

4.3 Evaluation Of Nearest Neighbor Re-Clustering Of Element Sections Corresponding To CAS Clustering

We clustered XML documents structurally and three clusters were produced (step 6 of Fig. 1). Then, we re-clustered concatenated element sections of XML documents of three primary clusters based on their content similarity using the matrix similarity. The clustering method in this step is the nearest neighbor that produced some subclusters, which are similar in two aspects of content and structure.

The Entropy of each subcluster and the total Entropy of subclusters for every three primary clusters was computed separately using Equation 4 and Equation 5. The abbreviations of A, B, and C are referred to as: the First K-means primary cluster, the Second K-means primary cluster, and The third K-means primary cluster in order.

The subcluster Entropy corresponding results of clusters A, B, and C are shown in Table 6, Table 7, and Table 8, respectively. The first column shows the cluster number of final produced clusters and the second column represents the cluster number of produced nearest neighbor clusters separately (i.e. A-1 is the first subcluster of the first primary K-means cluster or the first subcluster of cluster A). The third column expresses the size of each subcluster.

Table 6: Entropy results of subclusters of Nearest Neighbor clustering on the first K-means primary cluster (A)

Final cluster number	SubCluster number	Cluster size	Entropy
1	A-1	70	0.432
2	A-2	91	0.324
3	A-3	89	0.310
4	A-4	80	0.351
5	A-5	107	0.612
6	A-6	58	0.239
			Total Entropy= 0.3934

Table 7: Entropy results of subclusters of Nearest Neighbor clustering on the second K-means primary cluster (B)

Final cluster number	SubCluster number	Cluster size	Entropy
7	B-1	58	1.01

8	B-2	95	0.213
9	B-3	90	0.211
10	B-4	80	0.212
11	B-5	80	0.313
12	B-6	87	0.314
			Total Entropy= 0.3410

Table 8: Entropy results of subclusters of Nearest Neighbor clustering on the third K-means primary cluster (C)

Final cluster number	SubCluster number	Cluster size	Entropy
13	C-1	83	0.535
14	C-2	85	0.214
15	C-3	60	0.110
16	C-4	65	0.412
17	C-5	80	0.124
18	C-6	80	0.213
			Total Entropy= 0.2713

Eighteen subclusters were produced by applying the nearest neighbor clustering on element sections of three primary clusters. Overall, the best Entropy result is related to the C-3, while the worst is related to B-1. It can be inferred that in the C-3, most of the XML elements were located in a proper cluster, while in the B-1, many XML elements were clustered incorrectly. The best Entropy results are related to the subclusters of cluster C. However, due to the worst Entropy of its first subcluster, the total Entropy is near two others because the low quality of a cluster affects the quality of other clusters and the total quality. Smaller Entropy values indicate a better clustering solution. Thus, total Entropy results, corresponding to all clusters, are 0.337, meaning that the accuracy of subclusters is quite good; yet it can be further improved by methods such as feature selection, feature transformation and dimension reduction.

Comparing the results from the K-means clustering and nearest neighbor clustering reveals that K-means is much better than the nearest neighbor, showing higher-quality clusters. The differences can be explained as follows. Processing the text data is difficult due to its high dimension and difficulty of computing similarity between each pair of textual element sections. However, the extracted features of K-means clustering are mapped into binary values of "1" and "0", which are easier to process than the textual data. Such characteristic of data necessitates feature selection tasks to produce clusters with better quality. Eventually, 18 produced subclusters are considered together as their internal observations are similar based on both content and structure (CAS), while observations of different clusters are dissimilar. Thus, such clusters satisfy users more than the CO clusters.

4.4 Evaluating Efficiency of Structural Features of Proposed CAS Clustering VS. CO Clustering

In order to evaluate the proposed CAS clustering approach, the efficiency of structural features in the clustering of XML documents should be assessed. This evaluation was done locally by considering two different approaches. One approach is the proposed two-level CAS clustering of this study, and another is the common CO clustering. The results of the CO clustering approach were gained by applying the nearest neighbor clustering on the whole element sections. In this manner, K-means clustering was not primarily implemented on the XML documents; consequently, the structural aspects of XML documents were not considered and the XML element sections were clustered only based on their content similarity. Both results are shown in Table 9 to simplify the comparison of Entropy results.

Table 9: Entropy results of two-level CAS clustering vs. one-level CO clustering

Cluster No.	Entropy Result	
	Content Only (CO)	Content and Structural (CAS)
1	0.954	0.432
2	0.458	0.324
3	0.331	0.310
4	0.240	0.351
5	1.163	0.612
6	0.854	0.239
7	0.049	0.101
8	0.424	0.213
9	0.432	0.211
10	0.543	0.212
11	0.612	0.313
12	0.553	0.314
13	1.216	0.535
14	0.432	0.214
15	0.512	0.110
16	0.325	0.412
17	0.102	0.124
18	0.465	0.213
	Total = 0.5354	Total = 0.337134

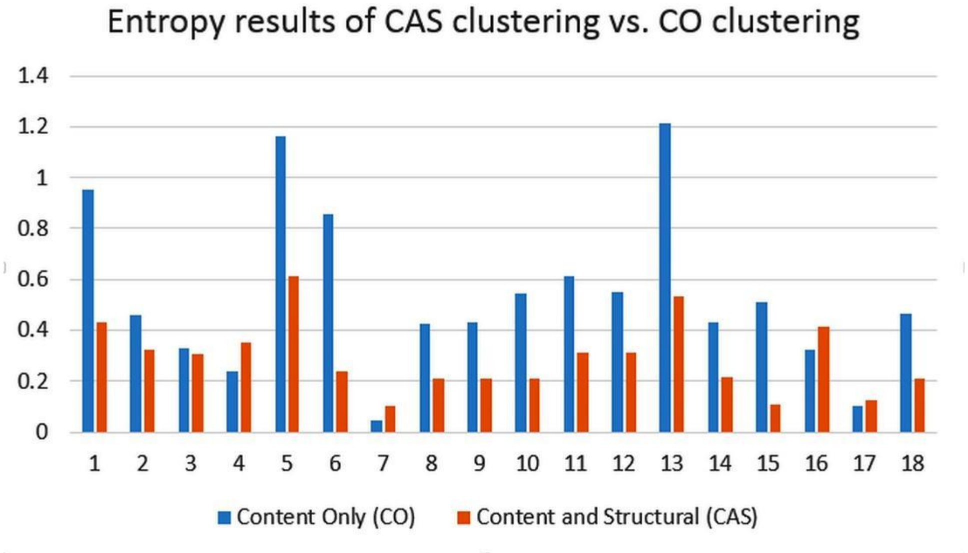

Fig. 6: Entropy result of two-level CAS clustering vs. one-level CO clustering

According to Table 9 and its completing Fig. 6, almost all Entropy results of 18 produced CO clusters are higher than the Entropy results of two-level CAS clustering. It can be perceived that the total Entropy results changed from 0.53 in the two-level CAS to 0.337 in the CO. This is the result of manipulation of structural features which, in turn come from the capability of XML structure compared with flat-text structure in general and applying an effective feature selection task in details of the clustering framework.

5.0 CONCLUSION

The main goal of this study was to consider the structural organization of text documents in the process of IR to achieve better precision using focused retrieval. It was meant to enable users to access elements from a known structure. Thus, a new clustering framework was proposed to improve the IR system in a DL by enhancing its data storage.

The proposed clustering approach of this study is a two-level approach that clusters textual element sections of XML documents, considering both the content and structural (CAS) similarity. Although the Entropy results of CAS clustering indicate that all the XML elements sections were not appropriately distributed, the improved results of the CAS clustering solution compared with the CO clustering denote the positive effect of structural features in better organization of a huge collection.

Since the gathered data set contains full-text scientific documents that are real documents from valid sources, it can be concluded that the proposed approach is practical in optimizing the organization of a huge corpus of documents in a DL. Thus, such a comprehensive clustering framework can be applied to the IR process. Indeed, clustering can be a practical solution to automatically organize a very large document collection into a minimal number of categories that should be searched to meet the user's query. With the decrease in the number of clusters needed to be searched, the efficiency of an IR system is improved, both in terms of search duration and quality of data retrieval.

Applying different dimension reduction strategies, such as feature selection and transformation methods in text preprocessing tasks, may significantly improve the results of clustering and the duration of such a heavy process is reduced. Another way to improve the Entropy results is by using ontology for semantic clustering to achieve high-quality clusters. Furthermore, in future work, we can increase the levels of hierarchical granularities as well as users' dynamic accessibility to the desired element.

ACKNOWLEDGMENTS

This research work has been supported by the University of Malaya RU Grant GPF098C-2020 and the Fundamental Research Grant Scheme FRGS/1/2020/ICT06/UM/02/1

REFERENCES

[1] A. Singhal, "Modern information retrieval: A brief overview" IEEE Data Eng. Bull, Vol. 24, No. 4, 2011, pp. 35-43.

[2] A. Algergawy, M. Mesiti, R. Nayak, and G. Saake, "XML data clustering: An overview", ACM Computing Surveys (CSUR), Vol. 43, No. 4, 2011, pp. 25.

[3] H. M. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, Intelligent search on XML data: applications, languages, models, implementations, and benchmarks. Springer Science & Business Media, 2003, pp. 59-75.

[4] E. Asghari, and M. KeyvanPour, "XML document clustering: techniques and challenges", Artificial Intelligence Review, Vol. 43, No. 3, 2015, pp. 417-436.

[5] A. Tagarelli, and S. Greco, "Semantic clustering of XML documents", ACM Transactions on Information Systems (TOIS), Vol. 28, No. 1, 2010, pp. 3.

[6] P. Dopichaj, "Content-oriented retrieval on document-centric XML", Dr. Hut Publisher, Munich, 2008.

[7] A. Trotman, "Wanted: Element retrieval users", In proceedings of the INEX 2005 on Element Retrieval Methodology, July 2005, pp. 63-69.

[8] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. England, Cambridge University Press, 2009, pp. 195-218.

[9] D. Magdaleno, I. E. Fuentes, and M. M. García", Clustering XML Documents Using Structure and Content based on a New Similarity Function OverallSimSUX", Computación y Sistemas, Vol. 19, No. 1, 2015, pp. 151-161.

[10] F. Dahak, M. Boughanem, and A. Balla, "A probabilistic model to exploit user expectations in XML information retrieval", Information Processing & Management, Vol. 53, No. 1, 2017, pp. 87-105.

[11] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh, "Finding an application-appropriate model for XML data warehouses", Information Systems, Vol. 35, No. 6, 2010, pp. 662-687.

[12] E. Wilde, and R. J. Glushko, "XML fever", Queue, Vol. 6, No. 6, 2008, pp. 46-53.

[13] N. Pharo, "The effect of granularity and order in XML element retrieval", Information Processing & Management, Vol. 44, No. 5, 2008, pp. 1732-1740.

[14] Z. Szlávik, A. Tombros, and M. Lalmas, "Summarisation of the logical structure of XML documents", Information Processing & Management, Vol. 48, No. 5, 2012, pp. 956-968

[15] T. Beckers, P. Bellot, G. Demartini, L. Denoyer, C. M. De Vries, A. Doucet, K. N. Fachry, N. Fuhr, P. Gallinari, and S. Geva, "Report on INEX 2009", ACM SIGIR Forum, New York, ACM, Vol. 44, No.1, 2009, pp. 38-57.

[16] R. Nayak, C. M. De Vries, S. Kutty, S. Geva, L. Denoyer, and P. Gallinari, "Overview of the INEX 2009 XML mining track: Clustering and classification of XML documents", In proceedings of 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, Berlin, Springer, Dec 2009, pp. 366-378.

[17] P. Arvola, J. Kekäläinen, and M. Junkkari, "Contextualization models for XML retrieval", Information Processing & Management, Vol. 47, No. 5, 2011, pp. 762-776.

[18] R. Dueñas-Fernández, J. D. Velásquez, and G. L'Huillier, "Detecting trends on the web: A multidisciplinary approach", Information Fusion, Vol. 20, 2014, pp. 129-135.

[19] M. Febrissy, and M. Nadif, "A consensus approach to improve nmf document clustering", In International symposium on Intelligent Data Analysis, Konstanz, Germany, Vol. 12080, Apr 2020, pp. 171-183.

[20] H. Schöning, "Tamino: a DBMS designed for XML", In proceedings of 17th International Conference on Data engineering, IEEE computer Society, 2001, pp. 149-154.

[21] A. Doucet, and M. Lehtonen, "Unsupervised classification of text-centric XML document collections", In International Workshop of the Initiative for the Evaluation of XML Retrieval, Berlin, springer, Dec 2006, pp. 497-509.

[22] M. Kc, M. Hagenbuchner, A. C. Tsoi, F. Scarselli, A. Sperduti, and M. Gori, "XML document mining using contextual self-organizing maps for structures", In International Workshop of the Initiative for the Evaluation of XML Retrieval, Berlin, Springer, Dec 2006, pp. 510-524.

[23] L. Denoyer, and P. Gallinari, "Report on the XML mining track at INEX 2007 categorization and clustering of XML documents", ACM SIGIR Forum, New York, ACM, Vol. 42, No.1, 2007, pp. 22-28.

[24] A. Al-Shammari, C. Liu, M. Naseriparsa, B. Q. Vo, T. Anwar, and R. Zhou, "A framework for clustering and dynamic maintenance of XML documents", In International Conference on Advanced Data Mining and Applications, Singapor, Springer, Nov 2017, pp. 399-412.

[25] R. Ibrahim, S. Zeebaree, and K. Jacksi, "Survey on semantic similarity based on document clustering", Adv. sci. technol. eng. syst. j, Vol. 4, No. 5, 2019, pp. 115-122.

[26] N. M. Salih, and K. Jacksi, "State of the art document clustering algorithms based on semantic similarity", Jurnal Informatika, Vol. 14, No. 2, 2020, pp. 58-75.

[27] A. Tagarelli, and S. Greco, "Toward Semantic XML Clustering", In Proceedings of the 2006 SIAM International conference on Data Mining, Washington, SIAM, 2006, pp. 188-199.

[28] T. Tran, R. Nayak, and P. Bruza, "Combining structure and content similarities for XML document clustering", In proceedings of the seventh Australasian Data Mining Conference, Australia, Vol. 87, 2008, pp. 219-225,

[29] I. Dongo, R. Ticona-Herrera, Y. Cadinale, and R. Guzmán, "Semantic Similarity of XML Documents Based on Structural and Content analysis", In proceedings of the 4th International Symposium on Computer science and Intelligent control, Newcastle, Nov 2020, pp. 1-9.

[30] K. Bessine, A. Nehar, H. Cherroun, and A. Moussaoui, "XCLSC: Structure and content-based clustering of XML documents", In 2015 12th International symposium on Programming and Systems (ISPS), Algeria, Apr 2015, pp. 1-7.

[31] N. G. Rezk, A. Sarhan, and A. Algergawy, "Clustering of XML documents based on structure and aggregated content", In 11th International Conference on Computer Engineering & Systems (ICCES), Egypt, Dec 2016, pp. 93-102.

[32] J. Tekli, and R. Chbeir, "A novel XML document structure comparison framework based-on sub-tree commonalities and label semantics", Journal of Web Semantics, Vol. 11, 2012, pp. 14-40.

[33] G. Costa, and R. Ortale, "A latent semantic approach to xml clustering by content and structure based on non-negative matrix factorization", In 12th International Conference on Machine Learning and Applications, Miami, Dec 2013, pp. 179-184.

[34] G. Costa, and R. Ortale, "Xml document co-clustering via non-negative matrix tri-factorization", In IEEE 26th International conference on Tools with Artificial Intelligence (ICTAI), Cyprus, Nov 2014, pp. 607-614.

[35] G. Costa, and R. Ortale, "Fully-automatic xml clustering by structure-constrained phrases", In IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Italy, Nov 2015, pp. 146-153.

[36] G. Costa, and R. Ortale, "XML Clustering by Structure-Constrained Phrases: A Fully-Automatic Approach Using Contextualized N-Grams", International Journal on Artificial Intelligence Tools, Vol. 26, No. 01, 2017.

[37] G. Costa, and R. Ortale, "Machine learning techniques for XML (co-) clustering by structure-constrained phrases", Information Retrieval Journal, Vol. 21, No. 1, Feb 2018, pp. 24-55.

[38] J. A. Hartigan, "Direct clustering of a data matrix", Journal of the american statistical association, Vol. 67, No. 337, 1972, pp. 123-129.

[39] D. Brzeziński, A. Leśniewska, T. Morzy, and M. Piernik, "XCleaner: A new method for clustering XML documents by structure", Control and Cybernetics, Vol. 40, No. 3, 2011, pp. 877-891.

[40] M. Piernik, D. Brzezinski, and T. Morzy, "Clustering XML documents by patterns", Knowledge and Information Systems, Vol. 46, No. 1, 2016, pp. 185-212.

[41] A. Muralidhar, and V. Pattabiraman, "An Efficient Association Rule Based Clustering of XML Documents", Procedia Computer Science, Vol. 50, Jan 2015, pp. 401-407.

[42] P. Bafna, D. Pramod, S. Shrwaikar, and A. Hassan, "Semantic key phrase-based model for document management", Benchmarking: An International Journal, Vol. 26, No. 6, 2019, pp.1709-1727.

[43] S. Jun, S.-S. Park, and D.-S. Jang, "Document clustering method using dimension reduction and support vector clustering to overcome sparseness", Expert Systems with Applications, Vol. 41, No. 7, 2014, pp. 3204-3212.

[44] C. Ding, and X. He, "K-means clustering via principal component analysis", In proceedings of the twenty first international conference on Machine Learning, Banf Alberta Canada, July 2004, pp. 29.

[45] C. H. Li, and S. C. Park, "An efficient document classification model using an improved back propagation neural network and singular value decomposition", Expert Systems with Applications, Vol. 36, No. 2, 2009, pp. 3208-3215.

[46] T. Ding, W. Li, and X. Li, "XML documents cluster research based on frequent subpatterns", In Sixth International Conference on Electronics and Information Engineering, Dalian, Vol. 9794, Dec 2015, pp. 97943A-97943A-5.

[47] E. L. Lydia, G. J. Moses, V. Varadarajan, F. Nonyelu, A. Maseleno, E. Perumal, and K. Shankar, "Clustering and indexing of multiple documents using feature extraction through apache hadoop on big data", Malaysian Journal of Computer Science, No. 1, 2020, pp. 108-123.

[48] L. Abualigah, A. H. Gandomi, M. A. Elaziz, A. G. Hussien, A. M. Khasawneh, M. Alshinwan, and E. H. Houssein, "Nature-inspired optimization algorithms for text document clustering—a comprehensive analysis", Algorithms, Vol. 13, No. 12, 2020, pp. 345.

[49] T. Bezdan, C. Stoean, A. A. Naamany, N. Bacanin, T. A. Rashid, M. Zivkovic, and K. Venkatachalam, "Hybrid fruit-fly optimization algorithm with k-means for text document clustering", Mathematics, Vol. 9, No. 16, 2021, pp. 1929.

[50] N. Samadi, "Clustering of XML documents for retrieval of heterogeneous digital libraries", Thesis in Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, 2013.

[51] M. Lalmas, "Xml retrieval (synthesis lectures on information concepts, retrieval, and services)", Morgan and Claypool, Jun 2009.

[52] A. Gupta, and S. A. Begum, "An empirical evaluation of the state of art feature selection methods for text categorization", International Journal of Scientific & Technology Research, Vol. 9, No. 2, 2020.

[53] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining", In LDV Forum, Vol. 20, No. 1, May 2005, pp. 19-62.

[54] M. F. Porter, "An algorithm for suffix stripping", Program, Vol. 14, No. 3, 1980, pp. 130-137.

[55] S. Zeebaree, R. R. Zebari, and K. Jacksi, "Performance analysis of IIS10. 0 and Apache2 Cluster-based Web Servers under SYN DDoS Attack", TEST Engineering & Management, Vol. 83, 2020, pp. 5854-5863.

[56] S. R. Zeebaree, K. Jacksi, and R. R. Zebari, "Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers", Indones. J. Electr. Eng. Comput. Sci, Vol. 19, No. 1, 2020, pp. 510-517.

[57] B. Larsen, A. Tombros, and S. Malik, "Is XML retrieval meaningful to users?: searcher preferences for full documents vs. elements", In proceedings of the 29th anuual international ACM SIGIR conference on Research and development in information retrieval, Seattle, Aug 2006, pp. 663-664.

[58] G. Salton, and M. J. McGill, Introduction to modern information retrieval. McGraw-Hill, Inc., 1983.

[59] W. Cohen, P. Ravikumar, and S. Fienberg, "A comparison of string metrics for matching names and records", In Kdd workshop on data cleaning and object consolidation, Aug 2003, pp. 73-78.

[60] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval", Information processing & management, Vol. 24, No. 5, 1988, pp. 513-523.

[61] P. Ganesan, H. Garcia-Molina, and J. Widom, "Exploiting hierarchical domain structure to compute similarity", ACM Transactions on Information Systems (TOIS), Vol. 21, No. 1, 2003, pp. 64-93.

[62] D. Lin, "An information-theoretic definition of similarity", In 15th ICML, Wisconsin, Vol. 98, Jul 1998, pp. 296-304.

[63] J. Tekli, R. Chbeir, and K. Yetongnon, "An overview on XML similarity: Background, current trends and future directions", Computer science review, Vol. 3, No. 3, 2009, pp. 151-173.

[64] C. C. Aggarwal, and C. Zhai, "A survey of text clustering algorithms", In Mining text data, Springer, 2012, pp. 77-128.