An Effective XML Documents Clustering Method Using Word Embeddings for Heterogeneous Collections

[1]B.A. Bodinga, [1]A. Roko, [1]A.B. Muhammad, [2]I. Saidu
[1]Department of Computer Science, Usmanu Danfodiyo University, Sokoto-Nigeria
[2]Department of ICT, Usmanu Danfodiyo University, Sokoto-Nigeria
*Corresponding Author: bello.bodinga@ududok.edu.ng, bello.bodinga@udusok.edu.ng

DOI: 10.56201/ijcsmt.v10.no2.2024.pg120.140

Abstract
As the size of XML repositories is growing, XML data management becomes challenging as how these documents can be stored and retrieved. One way of resolving such issues is to group the documents into clusters so that documents within the same cluster are more related than documents in different clusters. This became necessary in order to aid indexing and retrieval of XML documents. Traditional documents clustering methods represents documents with models that fails to consider the semantic relation between words. In this paper, WEClusterX is proposed to semantically cluster XML documents. The idea behind WEClusterX is to pinpoint which concept is represented by a particular context. Firstly, a pre-trained Bidirectional Encoder Representations from Transformers (BERT) is used to extract and cluster embeddings. Then, a Context-Document matrix is generated from the cluster of embeddings. Finally, clusters were formed using the famous k-means algorithm. The method combines the statistical importance of words with their contextualized representation in documents in order to forms meaningful clusters. The proposed WEClusterX is evaluated using extensive experiments. Experimental results have demonstrated that our proposed clustering solution achieved better performance in terms of purity and entropy.

Keywords: *XML document, Documents clustering, BERT, Embeddings, Heterogeneous documents.*

1. Introduction

Due to the simplicity and self – describing in the format of e**X**tendible **M**arkup **L**anguage (XML), the XML has become the main standard for document representation and exchange on the web [18]. As the number and size of XML repositories is growing, XML data management becomes challenging a show these documents can be stored and retrieved [14],[23],[27].. One way of resolving these challenges is to group the documents into clusters so that documents within the same cluster are more related than documents in different clusters [17], [36]. This became necessary in order to aid indexing and retrieval of XML documents. The key characteristic that distinguishes XML documents from traditional documents is their structured nature [18]. Conventionally, an XML document consists of structure (formed by tags and relationships between them) and content (that is the actual data stored in the document). This causes traditional

documents clustering techniques to become inappropriate for handling XML documents. This necessitated the proposal of several methods which can be used to cluster XML documents effectively. Several approaches to XML documents clustering have been developed over the years [2], [14], [16], [26-27] to enhance the performance of the XML retrieval system.

Over the years, XML document clustering has become a popular solution in XML Information Retrieval based on the idea that if a document is pertinent to a query, additional documents in that cluster can also relate to that query [27]. Grouping XML documents together helps improve data storage indexing, which will benefit the retrieval process [2], [14], [16], [26-27]. Only a small percentage of XML document clustering techniques focus solely on the content of the documents under consideration; the bulk arrange related documents into clusters based on both content and structure, or a combination of the two.

XML documents can emanate from the same source having the same structure (homogeneous) or from different sources having different structure (heterogeneous). Majority of the XML documents present high heterogeneity regarding their structure. Most of the existing clustering methods are applied on homogeneous collections. Hence, there is still a need of new approaches to manage and recognize similar information that consider the content and the semantics of the documents, besides the structure. Most current approaches semantically analyze the XML document content, regardless its structure or vice versa [14]. Nevertheless, these approaches give less importance to the semantic of the structure and the semantic context of terms. Thus, the terms that define the document structure also have semantic relation to the content considering the where the context terms appear. Managing vast amounts of heterogeneous documents for information searching can be improved with semantic analysis considering the context of terms. The structural and content similarities can be enriched with contextual analysis, such as the identification of real meanings of terms by working with the synonymy, polysemy, and relationship among them.

Several researchers have proposed different clustering methods to cluster XML documents. Some of these researchers considers only the content of the XML documents [16], [28], [31] thereby discarding the structural information of the documents since their similarity measure is based only on the content information stored in the XML documents. The Content-only methods aimed at creating textually similar groups of documents by utilizing existing text mining algorithms; other methods omit the content of a document and relies solely on its structure [1], [12], [29], [32].

Researchers in [16], [27-28], [30] utilizes both content and structure of the XML documents to form clusters that are similar in both content and structure.

Clustering methods that employs statistical measures suffers from the following limitations: (i) Polysemous ambiguities cannot be handled. (ii) relationship between words that shares the same context is mostly ignored since words are treated independently while utilizing simple frequency counts. Terms that have distinct meanings depending on the context are not handled. Since the meaning of a term can change depending on its relationship to another, the relationship between terms provides a new meaning to a comparable document. As a result, incorrect clusters are created. (iii) As the collection keeps growing, there are issues of dimensionality occurs.

In conclusion, none of the XML document clustering techniques now in use, as far as the researcher is aware, can handle these constraints. Consequently, a novel approach that can manage these

problems is required. This paper proposes WEClusterX - an effective XML documents clustering method using word embeddings in a heterogeneous environment.

1.2 Motivational scenario for context aware XML document clustering

XML keyword search suffers from several drawbacks especially in large scale XML data collection where data related to the user query appears in a small part (usually a fragment) of the whole XML document. Restricting the search to only parts of the data that might satisfy the user information need, the XML query processing can be conducted more efficiently because the search space is reduced by focusing on the document fragment potentially containing the search terms thereby avoiding unnecessary information during the query processing stage. A good solution is to consider clustering the whole XML document based on their common content, semantics, and structures [20]. Early researches [20], [28] has demonstrated that Executing the user query on the huge amount of XML documents is a time-consuming and error prone process. A nice solution is to first cluster together semantically and structurally similar XML documents. This is done by pinpointing which concept is represented by a context obtained from a document. Then, the user query is executed on one or more related clusters.

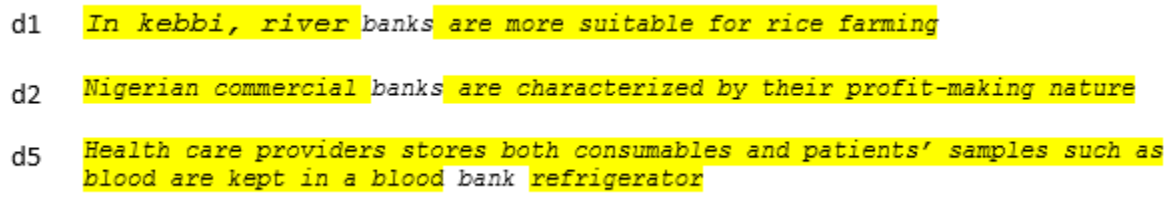

Figure 1: Sample text to explain context

For example, as seen in figure 1, Part of the text highlighted in yellow describes the context of the word 'bank' in the text. All the three documents belong to different concepts. The concept for d_1 is farming, d_2 is Accounting/Business and d_5 is medical laboratory concept.

To illustrate the role context plays in the process of grouping XML documents them by similarity, we consider a scenario in which eight XML documents (see figure 2) need to be grouped into certain clusters. Six documents have the same XML structure (title, abstract, author and keyword) and two documents has the same structure (book id, author, title, genre, price, publish date and description) with some keywords present in those documents can be used to group them. By a quick analysis, six of the eight XML documents have identical structure. Therefore, a structural comparison method would give wrong results since only two clusters is going to be formed.

Moreover, if a simple term occurrence method is used for the grouping, by considering only the keywords, another wrong result is obtained. For example, the keyword "bank" will be used to create cluster C_1 (d_1, d_3, d_5), keywords "XML and retrieval" will be used to create cluster C_2 (d_2, d_6) since the keywords appears in each document.

The remaining documents will form different clusters with C_3 (d_4), d_4 (d_7) and C_5 (d_8) respectively since the occurrences of keywords didn't match any document. The occurrences of the keyword "bank" used to form cluster C_1 are used in different contexts; hence these clustering solutions cannot work.

A correct grouping should be C_1 (d_1, d_7), C_2 (d_2, d_6), C_3 (d_3, d_8) and C_4 (d_4, d_5) respectively since those documents are semantically related.

In order to resolve this problem, the context of the terms used in the XML documents should be included when their content (occurrence of terms) and structure are analyzed.

<article>
<title>
An Improved Quelia Birds detection…. on Rice Farms with audio and visual sensors
</title>
<author> Ibrahim, Abdulsalam Magawata </author>
<Abstract>
Rice is one of the major foods …. In kebbi, river banks are more suitable for rice… the birds are more apparent in river bank areas according to studies by….
</Abstract>
<keywords> Rice, sensor, bank </keywords>
</article>

d1

<article>
<title>
An Effective PIA for XML document retrieval
</title>
<author> Roko, Abubakar </author>
<Abstract>
Predicates in XML are node that hold the text…. an effective search system is proposed… XML documents has …...
</Abstract>
<keywords> XML, predicate, Retrieval </keywords>
</article>

d2

<article>
<title>
An Enhance Document classification scheme in financial institution
</title>
<author> Roko, Abubakar </author>
<Abstract>
Nigerian commercial banks are characterized by…… several documents in the banks are related to credits, debits and loans…
</Abstract>
<keywords> Bank, credit, debit </keywords>
</article>

d3

<book id =”bk101”>
<author> Muhammad, Aminu Bui </author>
<title>Impact of ICT in controlled Hematology</title>
<genre> Compuer </Abstract>
<price> 44.95 </price>
<publish date> 2021-01-01 </publish date>
<description> an indepth look at usage of ICT Perfusion and montgomery</keywords>
</book>

d4

<article>
<title>
A Recommendation Engine for Storage facilities for Health care providers
</title>
<author> Almu, Abbba </author>
<Abstract>
Health care providers stores both consumables and patients’ samples such as blood….. the blood bank usually varies in--- depending on …
</Abstract>
<keywords> Consumables, storage, bank </keywords>
</article>

d5

<article>
<title>
A practical machine learning tools for XML data retrieval
</title>
<author> Muhammad, Aminu Bui </author>
<Abstract>
Several tools can be applied in …. WEKA can be used to …
</Abstract>
<keywords> XML, Machine Learning, Retrieval </keywords>
</article>

d6

<book id =”bk109”>
<author> Salihu, Saad Salisu</author>
<title> Crop Recommendation system </title>
<genre> Computer </Abstract>
<price> 44.95 </price>
<publish date> 2023-01-01 </publish date>
<description> crops such as rice prefer the coastal areas… the coastland mostly….. </description>
</book>

d7

<article>
<title>
LIA: a new indexing scheme for joint stock company records
</title>
<author> Mansur, Aminu </author>
<Abstract>
… most of the records in cash boxes are... the safe …
</Abstract>
<keywords> joint stock, cash box, records </keywords>
</article>

d8

Figure 2: Sample XML documents

2. Related Works

Clustering is used to explore the relationship among documents in a collection either from the same source (homogeneous) or from different sources (heterogeneous). Documents within a cluster are more similar to each other than documents belonging to a different cluster. XML document clustering can be performed by exploring their inherent features. This could be their content features, structural features or a combinational of both content and structural features.

2.1 Group I: Content Based approaches

Several methods to group XML documents based on their content similarity have been proposed over the years. These methods are mostly suitable for text – centric documents that have more content and very few structural information. Some notable method in this category include LAX [33], SLAX [34], 2-step method [12] and C^3M [5]. These methods consider only the content of the XML documents thereby discarding the structural information of the documents since their similarity measure is based only on the content information stored in the XML documents. The methods aimed at creating textually similar groups of documents by utilizing existing text mining algorithms. Due to wide prevalence of heterogeneous XML documents in real life datasets, content only approaches are insufficient to provide effective clustering solutions [35]. Another problem with these approaches is that the methods does not scale for large XML document collection due to the presence of large number of terms.

2.2 Group II: Structure Based approaches

Due to the structured nature of XML documents, many research efforts have been made to group XML documents based on their structural similarity. Some of the existing research works in this category include pioneer works that develop methods such as level ordered method [24], the two steps same pairs method [31], XClust [19], XSDCluster [22], famous CXPs [20], Dynamic clouds [32], Structure summaries method [12], XMine [22], XEdge [4], PCXSS [23], XProj [1] and CFSPC [31]. Other methods include XCLS [16], XCLS+ [3], XCleaner [16] and XPattern. These methods omit the content of a document and relies solely on its structure.

2.3 Group III: Content and Structure approaches.

Clustering methods that utilizes single feature focuses on either content or structural information contained in the XML documents. This tends to falsely group XML documents that are similar in both features. To correctly group XML documents, the clustering method shall combine both features to obtain optimal clusters. Some notable methods include cohesive subtree method [15], XCLSC [6], XEdge [26], A Non-negative Matrix Factorization technique called XC-NMF [7], XCO-CLUST [8] and XPart [9]. In [10-11], a method is proposed to cluster and maintain documents in a dynamic environment. The proposed framework reduces computational cost and tries to maintain existing clusters even if new documents arrive in a dynamic environment which reduces the computational cost. Researchers in [16] proposed a method called FEXC to cluster XML documents the edge sets. Researchers in [14] proposed remarkable framework named LSI* that enriches the content and structural similarity measure using semantic analysis using the

concepts of Singular Value Decomposition (SVD). In [27], a two-level clustering framework for clustering XML documents is proposed. This method uses the tf-idf and cosine similarity functions to to measure inter cluster similarity. The major limitation with this method is that it fails to put the actual semantics of terms in to consideration since its scoring function is only based on statistical measure. Words with different meanings in different context (polysemy) and words with same meaning (synonymy) are not handled. This led to the formation of inaccurate clusters.

3. Proposed Word Embedding Based XML documents clustering

In this paper, we proposed a BERT- based clustering method called WEClusterX to group XML documents not only based on structure and content information but also on semantic relatedness of terms. We define a new similarity function that can be used to cluster XML documents. Our approach extends the method proposed in [27] and establish a new similarity measure by handling term with context. BERT model has been successfully applied in clustering of plain text documents and achieve a good performance.

3.1 XML Documents Collection

In this paper, Real life heterogeneous XML documents collection were chosen. This collection consists of XML documents conforming to more than one structural definition. For the conduct of experiments, Niagara, Publication and DBLP datasets were used. The dataset is summarized in table 1. Detailed description of the dataset is also presented in table 3.1.

Table 1: A Summary of Document collection Statistics

XML document collection	Number of documents	Number of Classes	Collection Type
Niagara	496	23	Heterogeneous
DBLP	4,910	8	Heterogeneous
Publication	5,289	4	Heterogeneous

3.2 Architecture of *WEClusterX*

The proposed *WEClusterX* includes documents preprocessing and embeddings extraction module, clustering of embeddings module and generate clusters module. Figure 4.1 shows the architecture of the proposed clustering method. The *WEClusterX* method automatically clusters XML document collections based on context in addition to content and structural features. A cluster contains documents closely related in terms of concept. In the embedding module, the pre-trained language representation model BERT is used to generate contextualized sentence embeddings in the first step. We choose BERT base-sized model structure, which is a 12-layer bidirectional Transformer encoder consisting of the original implementation explained in the original paper. Next, the generated contextualized sentence embeddings are clustered in the second module while the third module group the XML documents into clusters. Detailed implementation is presented in section 3.2.

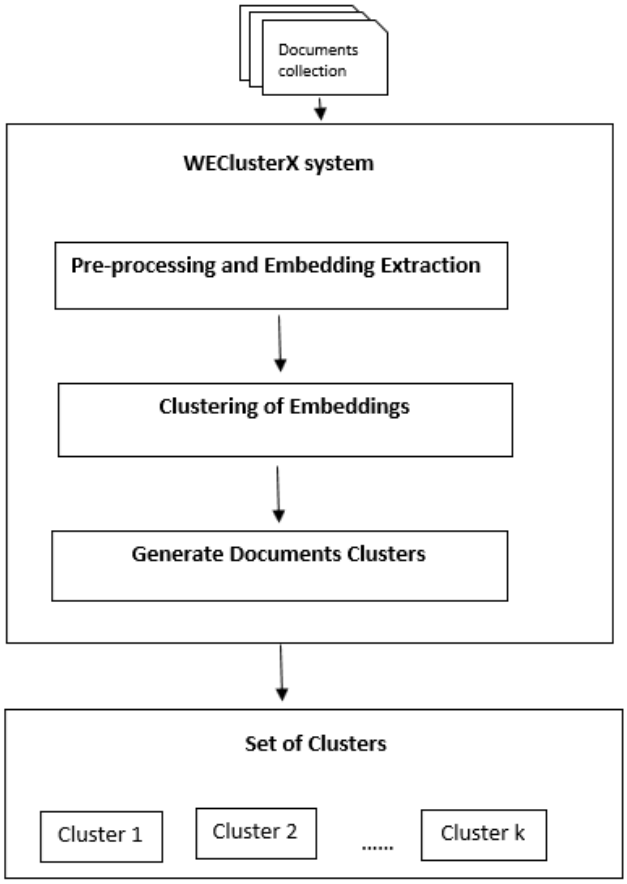

Figure 3: Architecture of the proposed method

3.3 Word Embedding-Based XML Documents Clustering method

This section presents the word embeddings based XML documents clustering solution. The section starts highlighting the role context plays in the clustering process in addition to the content and structural similarities of the XML documents. A detailed description of the proposed XML documents clustering method called WEClusterX is also presented.

A. Preliminaries

This section presents the word embeddings based XML documents clustering method that groups XML documents based on context.

As discussed in section two, the three groups of the existing XML documents clustering works will group the documents in to a single cluster. Our motivating scenario (see Figure 2) involves categorization of eight XML documents into certain groups. According to a similarity method followed by structure-based methods, the eight documents will be grouped together to form two clusters with C_1 (d_1, d_2, d_3, d_5, d_6, d_8) and C_2 (d_4, d_7) since six of the eight documents have identical structure. structure-based methods are not able to group the XML documents correctly. The same scenario when applied to content and structure methods also produces wrong clusters. For

example, using a term occurrence method for measuring content similarity for d_1, d_3 and d_5, the similarity result will group them into a single cluster, since the XML documents have the same frequency of terms (the keyword "bank" appears 1 time in each XML document). In this research, we proposed a method called *WEClusterX* to group XML documents not only based on structure and content information but also on context in which terms are used.

B. The WEClusterX clustering solution

In this sub section, a detailed description of the proposed XML documents clustering method called WEClusterX is given. WEClusterX combines the semantic advantages of the contextual word embeddings derived from the BERT model with statistical importance of words in documents obtained using TF_IDF scoring mechanism. The motivation behind this method is that TF_IDF scoring can capture the statistical importance of each word with respect to the document collection while BERT embeddings can capture context-based semantics of each word. To this end therefore, WEClusterX provides a unique way of combining the statistical and semantic features of the text.

As mentioned in the architecture, the proposed WEClusterX is described by the following modules.

I. Documents pre-processing and Embeddings Extraction

This is the first module of our clustering method where text pre-processing operations are applied such as stop words removal and stemming documents to their root words.

In the documents pre-processing stage, two features need to be extracted: the content and structure. To extract the features, SAX technology is used in this paper because it is faster than DOM parsing as it parses the elements in the XML document one by one starting from the root element. The content of the documents is pre-processed as follows: the text of the element nodes and attribute nodes are extracted. The text is tokenized by spaces and numbers and special characters are removed. Then common words known as stop words such as 'the', 'and', 'their' and the likes of them are removed. Then terms such as 'banking', banker' are stemmed to their root 'bank'. Additionally, terms whose length is less than three are removed.

Table 2 describes the pre-processes XML documents collection. The minimum and maximum level refers to number of levels in the hierarchical structure of the XML documents.

Table 2: Details of the pre-processed Heterogeneous Documents Collection

XML Document collection	No. of Internal Nodes	No. of Leaf Nodes	No. of attributes	Maximum Level	Minimum Level	No. of terms	No. of distinct terms
DBLP	9820	41196	11288	4	2	116960	22259
Publication	48908	122835	54805	6	3	532913	40588
Niagara	100682	383810	6067	16	2	865846	35826

From the pre-processed data in table 2, Niagara collection is large and more complex than DBLP and Publication collections. DBLP collection has a relatively small structure with maximum level of four. Based on the number of terms, on the average, each document in Niagara collection

contains 188 terms, each document in Publication collection contains 100 terms and each document in DBLP collection contains 23 terms.

In addition, all documents are prepared in a format suitable for processing with BERT. Every BERT encoder has a preprocessing model. It transforms raw text into the numeric input tensors that the encoder expects by using TensorFlow operators from the TF.text package. Different from pure Python preprocessing, these processes are integrated into a TensorFlow model for serving directly from text input. BERT is capable of producing case-sensitive embeddings for words, however, all the documents are converted in lower case for simplicity. As BERT finds contextually dependent embeddings, it takes a complete sentence as its input. Next, embeddings are extracted from the preprocessed collection.

For example, to cluster documents from heterogenous sources, one has to extract essential information from those documents and represent it in a form that will facilitate document comparison. At this stage however, the pre-processed collection is fed into the pre-trained BERT model. As a result of this, all words in the documents are converted into vectors of size 768 by applying $BERT_{base}$. Then, all embeddings that are not semantically important and do not play any role in discriminating the documents are removed. These include the embeddings corresponding to stop words, punctuations, and digits.

This module is implemented in algorithm 1. The algorithm takes the XML document collection, the pre-trained BERT model, list of stop words and punctuations as input and return list of word embeddings as output. The algorithm works as follows: Algorithm 1 split the XML document collection into sentences in line 2 and loops through the sentences from line 4 to 6. It takes each sentence and convert it to lower case and pass it to BERT model and obtain the corresponding word embeddings of the contents and stores the result in line 6. For example, figure 4 shows some sample sentences from our example XML document. The pseudocode of the algorithm is given in algorithm 1.

Algorithm 1: Pre-processing and extraction of BERT embeddings (from our documents collection)

Input: XML document collection --- dataset, pre-trained BERT model --- model, list of all stop words --- stop_words, list of all punctuations --- punc.
Output: list of word embeddings as result.

```
1. procedure PREPROCESS (dataset, model)
2. sentences = dataset.split('.')
3. results =[]
4.  for each sentence s in sentences do
5.      s = s.lower()
6.      results.append model(s)
7.   end for
8. r1 = stop_words
9. r2 = punc
10. r3 = digits
11. remove = r1 + r2 + r3
12. for each item in remove do
13.       remove stop_words, punc and digits.
14   end for
15.  return result
```

when these documents are passed as input to the algorithm, the algorithm loads a pre-trained bert_base_uncased model and apply get_embeddings method to extract the sentence embeddings. Figure 5 shows the extracted embeddings using sample document (figure 4).

docID	text
D1	In kebbi, river banks are more suitable for rice farming
D2	Predicates in XML are node that hold the text
D3	Nigerian commercial banks are characterized by their profit-making nature
D4	An in-depth look at usage of ICT in Perfusion and serum
D5	Health care providers stores both consumables and patients' samples such as blood are kept in a blood bank refrigerator
D6	Several tools can be applied in XML Information Retrieval
D7	Crops such as rice prefer the coastal areas
D8	most of the records in cash boxes are for credits and lending for reconciliation

Figure 4: Splitted sentences from example document

Due to the fact that BERT model is case sensitive, it produces embeddings for all contents in the document collection and some of these embeddings are unwanted. The algorithm removes unwanted embeddings in line 8 to 13 and then returns the list of actual word embeddings in line 15. Notice that in figure 4.3, for each sentence, a 768-dimensional vector is generated. Each dimension represents a virtual feature that captures a particular meaning (Devlin, 2019).

	generated_sentence	*Emb*
0	In kebbi, river banks are more suitable for rice farming	[5.273158, -2.0287377, -1.5211377, 1.1139345, ...
1	Predicates in XML are node that hold the text	[2.3629131, -0.5137803, 2.1055837, 0.8163596, ...
2	Nigerian commercial banks are characterized by their profit-making nature	[-0.00775963, 0.9391005, -2.2733262, 1.359431, ...
3	An in-depth look at usage of ICT in Perfusion and serum	[3.079915, -0.70477533, 1.2354763, -1.2202247, ...
4	Health care providers stores both consumables and patients' samples such as blood are kept in a blood bank refrigerator	[2.9549632, 0.7056699, 2.6203506, 0.35995768, ...
5	Several tools can be applied in XML Information Retrieval	[2.1529101, -0.5341803, 2.1251838, 1.2163596, ...
6	Crops such as rice prefer the coastal areas	[5.473128, -1.5287377, -1.5287377, 1.0139349, ...
7	most of the records in cash boxes are for credits and lending for reconciliation	[-0.01775461, 1.0391255, -2.0733161, 1.259201, ...

Figure 5: Extracted sentence embeddings

II. Clustering of Embeddings

The conversion of all the words into a numerical vector format makes it very easy and accurate to measure similarity between words/sentences.

BERT can capture semantic and contextual information from sentences rather than just looking at the word/tokens as done in Samadi and Ravana, 2023. For example, given the following sentences in figure 6, Samadi and Ravana, 2023 will put all the sentences in one cluster since it only uses term occurrences in forming clusters without performing any semantic comparison between word and its surrounding terms.

In kebbi, river banks are more suitable for rice farming

Nigerian commercial banks are characterized by their profit-making nature

Health care providers stores both consumables and patients' samples such as blood are kept in a blood bank refrigerator

Figure 6: Sample sentences

This kind of semantic comparison between words/tokens was not much accurate before the introduction of BERT. n number of clusters were identified from the sentence vectors in high 768-dimensional space. The primary purpose of these clusters was to identify similar sentences. The sentences in the same cluster contain the same context. Here, all the sentence vectors obtained from the previous module are arranged in the form of a matrix of dimension (n × 768). K-means clustering algorithm is applied to this matrix which results to the formation of clusters of words which we called concepts. Instead of using individual words as a vocabulary to represent any document, the idea is to use these contexts as the new vocabulary. The total size of the vocabulary is equal to the number of clusters denoted as k_{conc}. Then the first most significant turning point is

taken to be the number of clusters. As a result of the clustering of embeddings, the size of vocabulary gets reduced drastically from tens of thousands to less than a hundred.

Algorithm 2 generates set of concepts from collection, using number of concepts to find, list of documents and BERT embeddings for all documents as input. The algorithm works as follows: in line 3, the algorithm loop through the collection and for each document in the collection it assigns tf_idf score in line 4 and stores the tf-idf in line 5. In line 6, the algorithm iterates through the words in the document and stores the word content in the array called words_for_clustering in line 7 and stores the corresponding BERT embeddings in in an array called vec_for_clustering in line 8. The algorithm clusters the embeddings in line 11 and generate clusters of words in line 12 and append the result in line 13

Algorithm 2: Generating set of concepts from document collection

Input: number of concepts to find --- kconc, XML document collection --- dataset, BERT embeddings for all documents --- corpus_bert_embeds and number of iterations to be used in k-means ---t.
Output: Set of concepts as vocab_set.

```
1. procedure      Concept_Extraction(kconc,dataset, corpus_bert_embeds)
2. TF-IDF_corpus = []
3. vec_ for_clustering = []
4. word_ for_clustering = []
5. vocab_set = []
6. for each doc in corpus do
7. Assign doc_T F − I DF the TF-IDF scoring for each word in the document
and store it as a map between the word and its score.
8.        Add doc_T F − I DF to the list TF-IDF_corpus.
9.        for each word in doc do
10.       Add word to the single list word_ f or_clustering.
11. Add corresponding BERT embedding to the single list vec_ for_clustering.
12.       end for
13. end for
14.embed_categories = K-MEANSCLUSTERING (kconc, vec_ for_clustering,
t)
15. Generate cluster of words as cluster_words corresponding to clustering
achieved as embed_labels in the previous step.
16. Append cluster_words to vocab_set.
return vocab_set, TF-IDF_corpus
17.  end procedure
```

III. Generate Documents Clusters

This is the final module of our clustering solution. From the set of contexts obtained from the previous module, each document is now represented in terms of all the contexts. As a consequence, the entire collection is now represented in the form of a matrix called a Contexts - Document (CD) matrix. Each context is given a score in each document to indicate its degree of relevance to that document. The scoring mechanism for an i^{th} document d_i for j^{th} concept c_j is represented by CD_{ij} which is defined by the equation 1.

$$\mathrm{CD}_{ij} = \sum_{t=1}^{k} TF\_IDF(W_{jt}) \quad (1)$$

Where TF is given by

$$TF = freq\ (W_{jk}) \quad (2)$$

and IDF is defined by

$$IDF(W_{jk}) = (\log(\frac{|D| + 1}{doc\_count(W_{jk})+1}) + 1) \quad (3)$$

Therefore TF-IDF for all the k words is computed by multiplying equation 2 and 3 presented in equation 4.

$$TF - IDF(W_{jk}) = freq\ (W_{jk}) \ * \ (\log(\frac{|D| + 1}{doc\_count(W_{jk}) + 1}) + 1) \quad (4)$$

Where $|D|$ is the total number of documents in the documents collection D, freq(w_{jk}) is the frequency of word w_{jk} in document d_i and doc_count(w_{jk}) is the total number of documents that contain the word w_{jk} . The size of the matrix CD comes out to be (no. of documents x vocabulary size).

In this stage, the obtained knowledge during the clustering of embeddings stage is used to group the documents into clusters after the comparison is done in previous stage. This document clustering is performed by applying K-means clustering on the CD matrix (Na, Xumin and Yong, 2010). Because the number of features that are used to represent a document is drastically reduced, the k-means algorithm performs nicely on the input matrix. As a result of this phase, well-separated clusters of documents are achieved. The information about which pairs of documents are alike is usually stored in a similarity matrix, which contains data describing the distances between all documents in the dataset.

The actual clustering of the XML document collection is performed in algorithm 3. This algorithm takes sets of concepts and TF_IDF scores of all documents as input and produces clusters of documents as output, in line 2, the algorithm, initializes the document-concept matrix and loops through the documents collection in line 3 to 6 and for each document in collection and concept in vocab_set, the algorithm assign a score to each document based on the concept it represent using equation 4.1 and perform the clustering in line 8 using k-means algorithm (Na, Xumin and Yong, 2010).

Algorithm 3: Generating document clusters for WEClusterX

Input: Set of concepts --- vocab_set, TF-IDF scores for all documents --- TF-IDF_corpus .
Output: Clusters of documents.

```
1. procedure      DOCUMENTCLUSTERS(kconc, corpus, corpus_bert_embeds)
2. doc_concept = [][]
3. for each doc di in corpus do
4.     for each concept cj in vocab_set do
5.     Assign CDij a value k TF − IDF(wjk) using TF-IDF_corpus //using equation 1
6.  end for
7.       end for
8. Perform document clustering.
9: end procedure
```

4. Experiments

This section describes the experiments conducted to investigate performance of our proposed word embedding based XML documents clustering method.

4.1 Experimental Setup

This section presents the experimental setup used in this study, which consists of the experiment's environment and dataset. Python programming language is used for the implementation on windows 10 Professional 64-bit operating system runs on an Intel (R) Core i3 machine with 3.2.0 GHz processor and 8GB of RAM. Niagara, DBLP and Publication datasets were used to evaluate the clustering solution. Even though these XML documents collections are not extremely huge in term of number of documents, the collection has the following properties which make them good candidates for our experimental evaluations: i) the collections are heterogeneous ii) the distributions of the documents across classes are different and iii) the documents in the collection vary in both content and structure. Details of these datasets are presented in table 2.

4.2 Performance Metrics

In this paper, the quality of XML document 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. We used the Entropy and purity to evaluate clusters because this metrics are widely used in the evaluation of clustering approaches [27].

purity

Purity is a quantitative assessment of homogeneity of the content in a given cluster. Hence, purity measures the degree to which a cluster contains XML data primarily from one class. According to [37], the purity of cluster Ci is defined as:

$$\text{Pur}\ (\text{ci} = \frac{1}{\text{Ni}} \max\ (\text{N}_\text{i}^\text{r}) \qquad (5)$$

which is nothing more than the fraction of the overall cluster size (Ni) that represents the largest class of documents (N_i^r) assigned to that cluster. The overall purity of the clustering solution is obtained as a weighted sum of the individual cluster purities and it is:

$$\text{Purity} = \sum_{i=1}^{k} \frac{Ni}{N} \text{ pur (ci)} \quad (6)$$

The larger the values of purity, the better the clustering solution is.

Entropy

Entropy is a widely used measure for clustering solution quality, which measures how the various classes of the XML data are distributed within each cluster.

The entropy of a cluster Ci is defined as:

$$\text{Entr (Ci)} = \frac{1}{\log q} \sum_{i=1}^{q} \log \frac{N_i^r}{Ni} \quad (7)$$

where q is the number of classes in the XML dataset, and N_i^r is the number of XML data of the rth class that is assigned to the cluster ith. The entropy of the entire clustering solution is then defined to be the sum of the individual cluster entropies weighted according to the cluster size. That is,

$$\text{Entropy} = \sum_{i=1}^{k} \frac{Ni}{N} \text{ Entr (Ci)} \quad (8)$$

A perfect clustering solution will be the one that leads to clusters that contain documents from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution.

4.4 Experimental Results

This section describes the experiments conducted to investigate performance of our proposed word embedding based XML documents clustering method. We compare the performance of our method with two existing approaches. After executing the proposed approach, many important results are achieved. In this subsection, the results of WEClusterX in comparison with Samadi and Ravana (2023) are presented in detail in tables 3 and 4 and figures 7 and 8 respectively.

Table 3: Total Entropy Results

	Total Entropy Result	
Dataset	Samadi and Ravana (2023)	WEClusterX
Niagara	0.458	**0.232**
DBLP	0.264	**0.151**
Publication	**0.159**	0.243

Table 3 describes the total entropy results obtained for the three datasets. Best results are presented in bold.

Table 4: Total Purity Results

	Total Purity Result	
Dataset	Samadi and Ravana (2023)	WEClusterX
Niagara	0.856	**1**
DBLP	0.842	**0.97**
Publication	0.911	**0.961**

Table 4 describes the total purity results obtained for the three datasets. Best results are presented in bold.

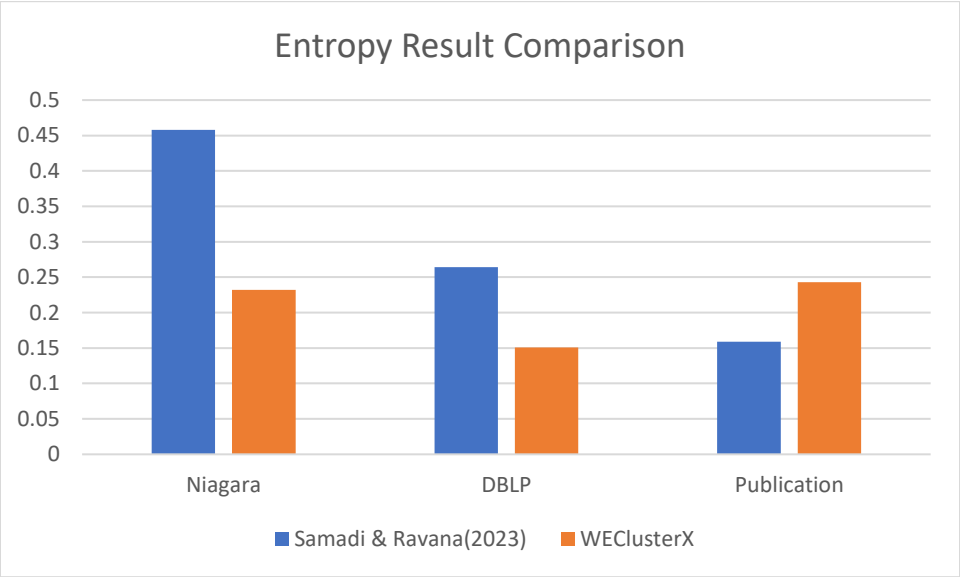

Figure 7: Total Entropy Result for the three datasets

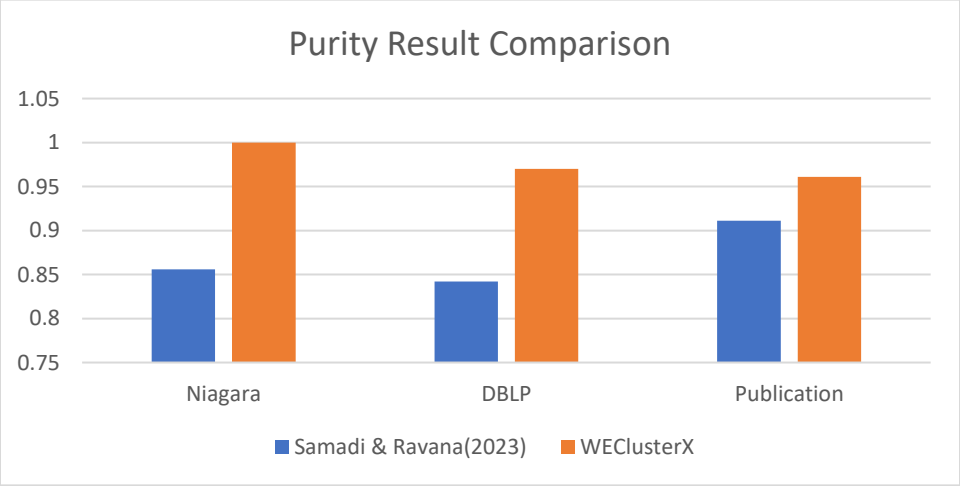

Figure 8: Total Purity Result for the three datasets

5. Result Analysis and Discussion

From the experiments, our clustering method achieves lower entropy values for Niagara and DBLP datasets. But for publication dataset it scores little higher value compared to [27]. This is probably due to the reason that publication dataset is small in size compared to Niagara and DBLP as it is relatively easier to find clusters in it. Best values are indicated in bold.

With regards to purity, the experimental results indicate that our clustering method achieves much more higher purity values for all the three datasets compared to [27]. Again, it is very clear that our clustering method outperforms the other technique except for one dataset. A visualization chart corresponding to these values are presented in Figure 6 and 7. It can be inferred from table 2 that for each of the datasets, there is a significant performance improvement.
It should also be noticed from the figure that as the size of the dataset grows, more improvement performance is taking place. This trend proves the efficiency of the proposed technique for large datasets. The reason behind these results can be attributed to the fact that word embeddings derived from the BERT model capture the semantics of the word and its context better. Other clustering techniques that are just based on a scoring mechanism like TF_IDF cannot capture the meaning of a word with respect to its context. The proposed technique combines the advantages of statistical scoring mechanisms like TF_IDF as well as the semantics of the word. Additionally, the clustering of word embeddings using K-means combines the words with similar contexts into a single group or a cluster. Hence, it reduces the dimensionality of the problem drastically i.e. from tens of thousands to less than a hundred. This enables the formation of more accurate clusters.

6. Conclusion and future work

In this paper, a new method, WEClusterX is presented, to address the problem of polysemous ambiguities in heterogeneous XML documents collection. A number of experiments were conducted to evaluate the performance of WEClusterX system with the benchmark. The results show that WEClusterX significantly outperforms Samadi and Ravana (2023) in terms of both purity and entropy. A promising development for the future improvements would be utilizing this concept in a dynamic environment where documents content and DTD may vary over time.

References

[1]. Aggarwal, C.C., Ta, N., Wang, J., Feng, J. and Zaki, M.J., (2007). XProj: A framework for projected structural clustering of XML documents. Proceedings of the 13th ACM SIGKDD International Conference on knowledge discovery and data mining, San Jose, Califomia, USA, pp 46-55

[2]. Aggarwal, C.C. and Zhai, C. (2012). A survey of text clustering Algorithms. In: Aggarwal, C.C. and Zhai, C. (eds) Mining Text data. Springer, Boston, MA. http://doi.org/10.1007/978-1-4614-3223-4_4.

[3]. Alishahi,M., Naghibzadeh, M., and Aski, B.S., (2010). Tag name structure-based clustering of XML documents. International Journal of Computer and Electrical Engeneerining, Vol. 2(1).

[4]. Antonellis, P., Makris, C., and Tsirakis, N., (2008). XEdge: clustering homogeneous and heterogeneous XML documents using edge summaries. Proceedings of the 2008 ACM Symposium on Applied Computing, New York, USA, pp. 1082-1088.

[5]. Altingövde, I.S., Atilgan, D., and Ulusoy, O., (2010). Exploiting Index Pruning Methods for Clustering XML Collections. Proceedings of the 8th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2009, Brisbane, Australia, December 7-9, 2009, pp. 379-386.

[6]. Bessine, K., Nehew, A., Cherroun, H., and Moussaoui, A., (2015). XCLSC: Structure and content-based clustering of XML documents. Proceedings of the 2015 12th International Symposium on Programming and Systems (ISPS), 28-30 April 2015.

[7]. Costa, G., and Ortale, R., (2013). A latent semantic approach to xml clustering by content and structure based on non-negative matrix factorization. Proceedings of the 12th International Conference on Machine Learning and Applications, Miami, pp. 179-184.

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

[9]. Costa, G., and Ortale, R., (2015). Fully-automatic xml clustering by structure-constrained phrases. Proceedings of the IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 146-153.

[10]. Costa, G., and Ortale, R., (2017). XML Clustering by Structure-Constrained Phrases: A Fully-Automatic Approach Using Contextualized N-Grams. International Journal on Artificial Intelligence Tools, Vol. 26(1).

[11]. Costa, G., and Ortale, R., (2018). Machine learning techniques for XML (co-) clustering by structure-constrained phrases. Information Retrieval Journal, Vol. 21(1), pp. 24-55.

[12]. Dalamagas, T., Cheng, T., Winkel, K.J., Sellis, T., (2006). A methodology for clustering XML documents by structure. Information Systems, vol. 31(3) pp. 187-228.

[13]. Devlin, J., Chang, M., Lee, K., and Toutanova, M., (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 4171–4186, Minneapolis, Minnesota.

[14]. Dongo, I., Ticona-Herrera, R., Cadinalle, Y., and Guzman, R., (2020). Semantic Similarity of XML Documents Based on Structural and Content Analysis. Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control, November 2020.

[15]. Greco, S., Gullo, F., Ponti, G., and Tagarelli, A., (2011). Collaborative clustering of XML documents. Journal of Computer and System Sciences, Vol. 77 (6) pp. 988-1008**.**

[16]. Jianwu, Y., Cheung, W.K., and Chen, X., (2005). Integrating element and term semantics for similarity-based XML document clustering. Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), Compiegne, France, 2005, pp. 222-228

[17]. Joulin, A., Edouard G., Piotr B.,& Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 427–431, Valencia, Spain.

[18]. Lalmas, M. (2009). XML RETRIEVAL. (G. Marchionini, Ed.). Morgan & Claypool http://doi.org/10.2200/S00203ED1V01Y200907ICR007

[19]. Lee, L.M., Yang, L.H., Hsu, W., and Yang, X., (2002). XClust: Clustering XML schemas for effective integration. Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 4-9, 2002.

[20]. Leung, H.P., Chung, F.L., Chan, S.C.F., and Luk, R. (2005). XML document clustering using common XPath. Proceedings of the 2005 International Workshop on Challenges in Web Information Retrieval and Integration (WIRI 2005), 8-9 April 2005, Tokyo, Japan

[21]. Mikolov, T., Sutskever, I., and Chen, K., et al. (2013) Distributed Representations of Words and Phrases and Their Compositionality. Proc of the 26th International Conference on Neural Information Processing Systems, Curran Associates Inc., USA, 3111-3119.

[22]. Nayak, R.,and Xia, F.B.,(2004) Automatic Integration of Heterogenous XML-schemas. In Kotsis, G, Taniar, D, Bressan, S, & Ibrahim, I K (Eds.) The Sixth International Conference on Information Integration and Web-based Applications and Services. Oesterreichische Computer Gesellschaft, Bandung, Indonesia, pp. 427-436.

[23]. Nayak, R., and Tran, T., (2007). A progressive clustering algorithm to group XML data by structural and semantic similarity. International Journal of Pattern Recognition and Artificial Intelligence. Vol. 21(4). Pp. 723-743.

[24]. Nearman, A., and Jagadish, H.V., (2002). Evaluating Structural Similarity in XML Documents. Proceedings of the 5th international conference on computational science (ICCS), Wisconsin, USA.

[25]. Pennington,J., Socher,R., & Manning C.(2014). GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[26]. Rezk, N.G., Sarhan,A., and Algergawy, A., (016). Clustering of XML documents based on structure and aggregated content, Proceedings of the 11th International Conference on Computer Engineering & Systems (ICCES), Egypt, pp. 93-102.

[27]. Samadi, N., and Ravana, S.D., (2023). XML Clustering Framework based on document content and structure in a heterogenous digital library. Malaysian Journal of Computer Science, Vol. 36 (3), pp. 124-147.

[28]. Tagarelli, A., amnd Greco, S., (2006). Towards Semantic XML Clustering. In the proceedings of the SIAM International conference on data mining, pp. 188-198.

[29]. Tran, T., Nayak, R. (2007). Evaluating the Performance of XML Document Clustering by Structure Only. In: Fuhr, N., Lalmas, M., Trotman, A. (eds) Comparative Evaluation of XML Information Retrieval Systems. INEX 2006. Lecture Notes in Computer Science, vol 4518. Springer, Berlin, Heidelberg.

[30]. Tran, T., Nayak, R., Bruza, P. (2008). Document Clustering Using Incremental and Pairwise Approaches. In: Fuhr, N., Kamps, J., Lalmas, M., Trotman, A. (eds) Focused Access to XML Documents. INEX 2007. Lecture Notes in Computer Science, vol 4862. Springer, Berlin, Heidelberg.

[31]. Tran, T., Kutty, S., Nayak, R. (2009). Utilizing the Structure and Content Information for XML Document Clustering. In: Geva, S., Kamps, J., Trotman, A. (eds) Advances in Focused Retrieval. INEX 2008. Lecture Notes in Computer Science, vol 5631., pp. 460-468, Springer, Berlin, Heidelberg

[32]. Vercoustre, AM., Fegas, M., Gul, S., Lechevallier, Y. (2006). A Flexible Structured-Based Representation for XML Document Mining. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds) Advances in XML Information Retrieval and Evaluation. INEX 2005. Lecture Notes in Computer Science, vol 3977. Springer, Berlin, Heidelberg.

[33]. Wenxin, L., and Haruo, Y., (2005). LAX: An Efficient Approximate XML Join Based on Clustered Leaf Nodes for XML Data Integration, Proc. BNCOD 2005, Springer LNCS 3567, pp.82– 97.

[34]. Wenxin, L., and Haruo, Y., (2006). SLAX: An improved leaf-clustering based approximate XML join algorithm for integrating XML data at subtree classes. Information and Media Technologies 1, 2 (2006), 918–928.

[35]. Yao, J., and Zerida, N., (2007). Rare patterns to improve path-based clustering of wikipedia articles. Proceedings of the Sixth Annual Workshop of the Initiative for the Evaluation of XML retrieval (INEX'07), 2007, Germany. pp.224-231.

[36]. Yang, J., Cheung, W.K., and Chen, X., (2005). Learning the kernel matrix for XML document clustering. Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 353-358

[37]. Zhao, Y., Karypis, G. (2005). Data clustering in life sciences. Mol Biotechnol Vol. **31**, pp. 55 -80. https://doi.org/10.1385/MB:31:1:055