Book Chapter

A Novel Method for Transforming XML Documents to Time Series and Clustering Them Based on Delaunay Triangulation

Narges Shafieian*

Department of Computer Engineering, Zanjan Branch, Islamic Azad University, Iran

***Corresponding Author:** Narges Shafieian, Department of Computer Engineering, Zanjan Branch, Islamic Azad University, Zanjan, Iran

Published **January 10, 2020**

This Book Chapter is a republication of an article published by Narges Shafieian, et al. at Applied Mathematics in June 2015. (Shafieian, N. (2015) A Novel Method for Transforming XML Documents to Time Series and Clustering Them Based on Delaunay Triangulation. Applied Mathematics, 6, 1076-1085. doi: 10.4236/am.2015.66098.)

How to cite this book chapter: Narges Shafieian. A Novel Method for Transforming XML Documents to Time Series and Clustering Them Based on Delaunay Triangulation. In: Prime Archives in Applied Mathematics. Kyandoghere Kyamakya, editor. Hyderabad, India: Vide Leaf. 2020.

Acknowledgements: This work is a revised, extended version of the paper that appeared in [6,7]. The authors are very grateful to Michael Thomas Flanagan for providing Java Scientific

Library proposed in [25]. This work was exploited other standard java library like SAX, moreover, applied CGAL and also boost library for coding C++ part of the program.

Abstract

Nowadays exchanging data in XML format become more popular and have widespread application, Because of simple Maintenance and transferring nature of XML documents. So accelerate searching within such a documents ensure search engine's efficiency. In this paper we propose a technique for detecting the similarity in the structure of XML documents. The technique is based on the idea of representing the structure of an XML document as a time series in which each occurrence of a tag corresponds to a given impulse. So we could use Discrete Fourier Transform as a simple method to analyzing these signals in frequency domain and make similarity matrices though a kind of distance measurement, in order to group them into clusters. We exploited Delaunay Triangulation as a clustering method to cluster the d-dimension points of documents.

Keywords

XML Mining; Document Clustering; XML Clustering; Schema Matching; Similarity Measures; Delaunay Triangulation; Cluster

Introduction

The main idea of this method is based on structure of XML documents, means tags and position of elements in XML tree's hierarchy. So content of documents is not important, in other words, it may exists two documents with completely similar structure is related to completely different content and inverse.

Input of our method implementation is set of documents and output is clustering these documents into various clusters, and if clustering perform suitably, the documents in each cluster have the same structure and documents are belongs to different clusters have less structural similarity. The main contribution of our approach is these steps:

- Mapping each documents to a time series;
- Getting DFTs and transforming each time series from time domain to frequency domain;
- Mapping the signals related to each documents to a point in d-dimensional space;
- Triangulation of points related to documents;
- Clustering documents based on their triangulation.

For analyze accuracy of clustering, we use external metric. In analyzing based on external metrics, we would evaluate other clustering strategy in front of our proposed clustering method. Having more match between clustering metric and other clustering, clustering may have more precision too. We use two external metrics names F-Measure and Purity as evaluator of our method. More information about this method is mentioned in [22].

The corpus of documents for evaluating this method is a standard corpus, which a part of that is applied. This corpus has clustering metric itself which we used it as a comparison versus our external metrics. This corpus could download from [20] and is defined in [22].

The rest of the paper is organized as follows: In Section 2, we present some information about common methods for detecting similarities and clustering documents, section 3, expressed implementations requirements and developing environment. Section 4, illustrates how the structure of an XML document can be encoded into a time series, mapped to d-dimension space, triangulate and finally clustered, then, presents some methods for accomplishing such tasks. Section 5, describes several experiments we performed, on encyclopedia data set, to validate our approach. We sketch some issues which could be faced in future work on this topic, and conclude the paper in section 6.

Related Work Summary

Several methods for detecting the similarity of XML documents have been recently proposed, that is based on the concept of edit distance and use graph-matching algorithms to calculate a (minimum cost) edit script capable of transforming a document into another. Most of these techniques are computationally expensive, i.e. at least O (N3), where N is the number of element of the two documents. However, all of them are concerned with the detection of changes occurring in XML documents rather than comparing them on the basis of their structural similarity. Some approaches have a technique for measuring the similarity of a document versus a DTD is introduced. This technique exploits a graph-matching algorithm, which associates elements in the document with element in the DTD. This approach does not seem to be directly applicable to cluster documents without any knowledge about their DTDs, and is not able to point out dissimilarities among documents referring to the same DTD [13].

Indeed, we propose to represent the structure of an XML document as a time series, where each tag occurrence corresponds to an impulse. By analyzing the frequencies of the Fourier Transform of such series, we can state the degree of (structural) similarity between documents. As a matter of fact, the exploitation of the Fourier transform to check similarities among time series is not completely new [13] and has been proven successful. The main contribution of our approach is the systematic development of an encoding scheme for XML documents, in a way that makes the use of the Fourier Transform extremely profitable.

Hence, after detection of documents similarity we could group documents into different clusters, which intensively accelerate search engine motors. Particularly, XML document clustering

Algorithms divided into two groups:

- Pair wise methods
- Incremental methods

Pair wise based algorithms are more common which first create a similarity matrix for each pair of documents. This matrix is initialized by a criterion for measuring similarity between two documents. Finally, after completing the matrix we can use a general clustering algorithm such as K-means to locate a document in its proper cluster. In this paper we applied a new clustering method named Delaunay triangulation, which is used for clustering video frames. In contrast to many of the other clustering techniques, the Delaunay clustering algorithm is fully automatic with no user specified parameters [24].

Implements Requirements and Performing

Our proposed method developed by java and in order to run, need JRE or JDK (6 versions). Development environment is Eclipse (and using SAX, Flanagan library to implement some part of the program).

In one phase of project we need to triangulate some points in d-dimensional space, with Delaunay method. In consideration of large size of d (more than 3), this function extremely reduce efficiency. In order to increase efficiency, we use C++ language and CGAL library [23] for implementing this phase. This program is written and compile independently and it just triangulate points.

Clustering Phases

Mapping each Documents to a Time Series

In this phase XML documents parsed one by one and a time series produced, it means that a set of numbers which are in time ordered, produced. For producing time series a special coding function applied which is effectively reflect documents structure, means tags and hierarchal structure of XML tree. The coding function has two parts. One of them is local and gives a unique identity to each tag. In order to make this identity for each tag we use a random order. With every visit of a new tag, the tag was added to a data base and a unused identity assigned to it. From now to, with visit of that tag, this local code was looked up in data base and being used.

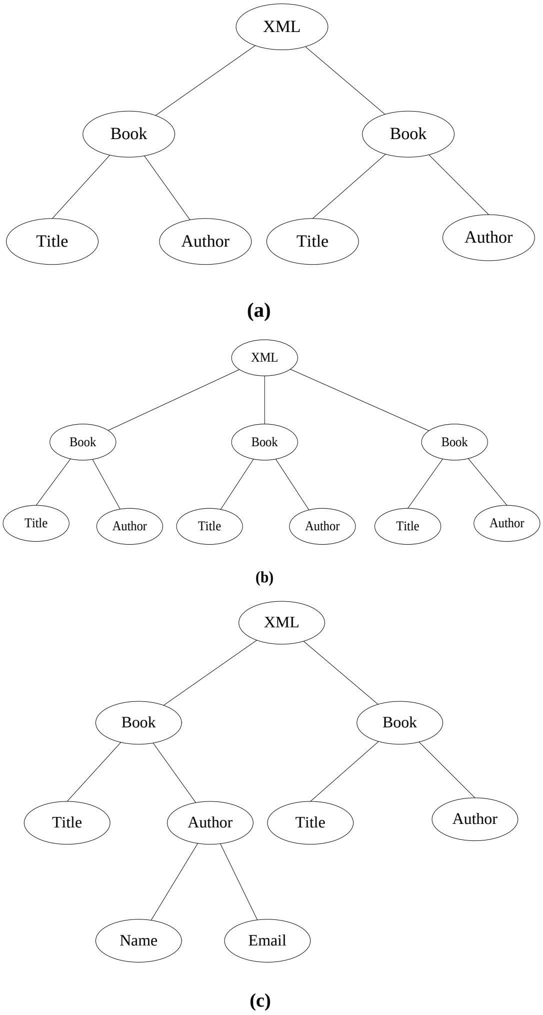
In addition, local coding based on tags, we use a un local coding function which, take attention to position of tags in hierarchy structure. This unlocal coding, assign a special weight to each tags based on its depth and hierarchy structure. Detailed information about labeling documents are find in [13]. Mapping documents to time series are done as below definition.

Definition 1: Let D be a set of XML documents, d a document in D with sk(d) $= [t_0, \dots, t_n\ ]$ and γ a tag encoding function for D. Moreover, let maxdepth(D) represent the maximum depth of any document in D, B a fixed value and $nest_d(t\ \ )$ the set of tag instances associated with the ancestors of the element with tag instance t. A multilevel encoding of d(mlemc(d)) is asequence$[S_0, S_1, \ldots, S_n]$, where:

$$S_i = \gamma(t_i) \times B^{maxdepth(D)-l_{t_i}} + \sum_{t_j \in nest_d(t_i)} \gamma(t_j) \times B^{maxdepth(D)-l_{t_j}} \quad (1)$$

We usually set B as the number of distinct symbols encoded by _ (e.g., B = $|tnames(D)\ | + 1$in the case of invariantγ_d). In this way, we avoid “mixing” the contributions of different nesting levels and can reconstruct the path from the root to any tag by only considering the corresponding value in the encoded sequence. In fact, the summation on the right-hand side of the above formula can be interpreted as the integer whose B-base representation is the sequence of the tag codes in$\{\gamma(t_j) | t_j \epsilon nest_d(t_i)\}$, ordered by increasing nesting levels of the corresponding tags. Notice that such a property is stronger than WSL, and is not mandatory for guaranteeing injectivity in the encoding function.

For example in the documents below, we could realized that (a) and (b) documents are more similar to document (c). In order to approve it via their transferred format, we should notice the next phase.



(a)

(b)

(c)

Figure 1: (a) book1 and (b) book2 have the same elements, but with different cardinality. By contrast, (c) book3 induces a different structure for the author element

Getting Discrete Fourier Transform (DFT) and transferring each time series from time domain to frequency domain

Time series produced in previous phase are in time domain means that these series reveal tags with its special structure, during time domain. The Length of these signals is different and comparison of them is difficult. In order to capture similarities and structural differences, signals transfer from time domain to frequency domain. So we could compare two signals magnitudes in specific frequency. This comparison reflected in structural differences between documents.

Consider Figure 2, representing the documents of Figure 1. Observe that all the signals have different shapes. Notwithstanding, the difference among the signals can be summarized as follows:

- Each book element is associated with a unique subsequence within the signals associated with book1 and book2. Nevertheless, the subsequences number's occurrences are different.

- Book3 has two different subsequences associated with the book elements. Moreover, the first subsequence is different from the ones in book1 and book2.

A comparison in the time domain (accomplished using the time-warping distance) will result in a higher similarity between book1 and book3 than between book1 and book2. Nevertheless, each different subsequence triggers a different contribution in the frequency domain, thus allowing for detecting the above described dissimilarities. To better understand how the differences between two documents reflect on the frequency spectra of their associated encodings, we can always consider these differences separately and exploit the linearity property of the Fourier transform.

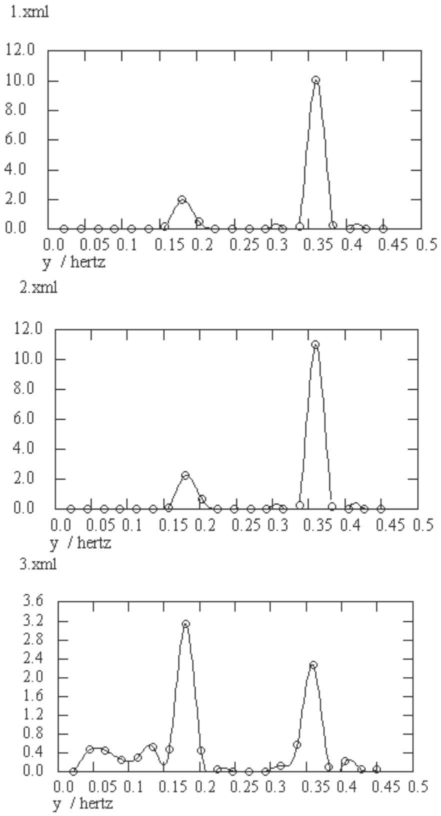
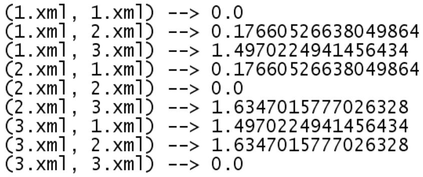


Figure 2: nonzero frequency components of the book1, book2, and book3 documents in Fig. 1,Discrete Fourier Transform

Definition 2: Let d1, d2 be two XML documents, and enc a document encoding function, such that $h_1 = enc(d_1)$ and $h_2 = enc(d_2)$. Let DFT be the Discrete Fourier Transform of the (normalized) signals. We define the Discrete Fourier Transform distance of the documents as the approximation of the difference of the magnitudes of the DFT of the two encoded documents:

dist (d_1, d_2) = $\left(\sum_{k=1}^{M/2}\left(\left|[D\tilde{F}T(h_1)](k)\right| - \left|[D\tilde{F}T(h_2)](k)\right|\right)^2\right)^{\frac{1}{2}}$ (2)

Where $D\tilde{F}T$ is an interpolation of DFT to the frequencies appearing in both d_1 and d_2, and M is the total number of points appearing in the interpolation, i.e., M = N_{d_i}+ $N_{d_j} - 1$.



```
(1.xml, 1.xml) --> 0.0
(1.xml, 2.xml) --> 0.17660526638049864
(1.xml, 3.xml) --> 1.4970224941456434
(2.xml, 1.xml) --> 0.17660526638049864
(2.xml, 2.xml) --> 0.0
(2.xml, 3.xml) --> 1.6347015777026328
(3.xml, 1.xml) --> 1.4970224941456434
(3.xml, 2.xml) --> 1.6347015777026328
(3.xml, 3.xml) --> 0.0
```

Figure 3: similarity matrix correspond to (a), (b) , (c) documents

Hence, produced frequency signals, completely present documents structural differences, which is turned out in similarity matrix, too. Means that bigger value of the cell, related to documents, reveal more distance and also lower similarity.

In order to transferring time series from time domain to frequency domain, Discrete Fourier Transform (DFT) has exploited. We use JAVA library to perform this transferring. This library using FFT algorithm which have suitable time cost too. For more information refer to [13].

Mapping Signal Corresponding to Each Document to a Point in D-dimensional Space

Finally we could sampling the signal consequence from previous phase and make a discrete signal. If sampling was done for the same frequency and signals magnitude compared in these positions, documents similarity can estimated. More sampling conclude to more accurate comparison and, in other hand, time cost of other remain calculation was increased. Thus, choosing a degree for sampling, have intensive influence to clustering efficiency and accuracy, so is a trade-off.

After this phase, each document mapped to signal point in d-dimension space, which d is the size of applied sampling. Now there are some points in d-dimension, which should be clustered. In other applications, we could map component of videos, images, sounds and etc. to points and use a clustering method for them too. Afterwards, we could compare this clustering method with others. In other word, we could give these points to them and compare the results. Each point can be as a feature vector.

Triangulate Points Corresponding Documents

From this phase, points correspond to documents is in d-dimension, triangulation was exploited because of following reasons: inside of produced triangles was no point or points. It means that each point is at the corner of one or more triangles but not in the any triangle. This good feature is desirable for high efficiency clustering which, is defined in the following part.

Notice that triangle are not only in only two-dimension and be any shape like pyramid and etc in higher dimensions. In d-dimension space each triangle made up d+1 corner. Consequence of this phase is a graph which vertex are points and its edges are sides of triangles produced from this triangulation method. The graph is saved as a file till, clustering algorithm used it. Triangulation in incremental way was implemented by CGAL library, which is complicated and when the dimension be higher, it’s done very slowly. Other triangulation methods exist for Delaunay but their problem was dimensions, too. Triangulation was done as below definitions.

The formal definitions for the mean edge length and local standard deviation for each data point follows from Definitions 3 and 4.

Definition 3: The mean length of edges incident to each point *pi* is denoted by Local_Mean_Length(*pi*) and is defined as

$$Local\_Mean\_Length(p_i) = \frac{1}{p(p_i)} \sum_{j=1}^{d(p_i)} |e_j| \qquad (3)$$

where *d(pi)* denotes to the number of Delaunay edges incident to *pi* and |*e j* | denotes to the length of Delaunay edges incident to *pi* .

Definition 4: The local standard deviation of the length of the edges incident to *pi* is denoted by Local_Dev(*pi*) and is defined as

$$Local\_Dev(\, p_i)= \sqrt{\frac{1}{d(\, p_i)}\sum_{j=1}^{d(\, p_i)}(Local\_Mean\_Length(p_i) - |\,\mathrm{e_j}|)^2} \quad (4)$$

To incorporate both global and local effects, we take the average of local standard deviation of the edges at all points in the Delaunay diagram as a global length standard deviation as defined in Definition 5.

Definition 5: The mean of the local standard deviation of all edges is denoted by Global_Dev(P) and is defined as

$$Global\_Dev(P) = \frac{1}{N}\sum_{i=1}^{N} Local\_Dev(p_i) \quad (5)$$

Where *N* is the number of total points and *P*is the set of the points.

All edges that are longer than the local mean length plus global standard deviation are classified as inter-edges (Definition 7) and form the separating edge between clusters. The formal definition for short and separating edges in terms of mean edge length of a point and mean standard deviation are captured in Definitions 4 and 5 below.

Definition 6: A short edge (*intra-cluster edge*) is denoted by Short_Edge(*pi*) and is defined as

$$Short\_Edge(p_i)=\{e_j\,\|e_j| < Local\_Mean\_Length(p_i) - Global\_Dev(P)\} \quad (6)$$

Definition 7: A Separating edge (*inter-cluster edge*) is denoted by Separating_Edge(*pi*) and is defined as

$$Seprating\_Edge(p_i) = \{e_j\|e_j| > Local\_Mean\_Length(p_i) - Global\_Dev(P)\} \quad (7)$$

Delaunay Triangulation can be done effectively in $O(n \log n)$ time and the identification of inter and intra edges can be done in $O(n)$ time where n is the number of documents processed. For detailed information refer to [24].

Clustering Documents Based on their Triangulation

The triangles obtained from previous phase are applied for clustering. The triangulation method has a feature that, having no point inside of any triangles, was guaranteed. With this key feature, analyzing every sides of any triangles to recognizing nearest and farthest points, could be possible.

Hence, we could disconnect links between points of graph by deleting fairly big edges. Finally, Graph was divided into some individual sections. Each of them made a new cluster. recognizing fairly big edges, was done in the way was illustrated in [24].

We use a parameter for control deleting bigger edges. This parameter that named clustering factor, demonstrating what rate of edges deviations expressed the edges should be deleting. This way is fairly complicated for, length mean of every edges end up to each vertex, their standards deviations and average of all standard deviation, should be computed.

Finally some file produced which, each of them contain some documents, were inside of a cluster. With higher value of clustering parameter, number of produced cluster became bigger. This parameter should choose in the way that suitable number of clusters been produced. We could do clustering more times for each set of XML documents to find its suitable clustering factor value.

Clustering Evaluation’S Parameters and Notifications

As mentioned before, for evaluating accuracy of proposed method, we use two parameters named Purity and F-Measure. These two parameters computation method was illustrated in

[22]. For computing the parameter which was in external type, a clustering metric was needed. Corpus of the used data set has this metric, this corpus explained in [21] as this:
Closer value of these parameters to 100 percent means clustering accuracy rate was higher, and in verse.

Clustering a set of documents with proposed method expressed that, these parameter isn't high enough. Mean that, this kind of clustering method suitable for clustering the data set.

Three parameters have influenced to clustering efficiency and accuracy, involved:

- Number of sampled points from documents frequency: more number of points is sampling from documents frequencies (produced diagram of transferring time series from time domain to frequency domain), made documents comparisons better. This parameter expressed the points' space dimensions. For intensive inefficiency triangulation in higher dimensions, increasing number of sampling made clustering run slowly, which exclude to lower performance in triangulation and clustering, too.
- Clustering parameter: higher value of this parameter, made number of produced clusters higher. This parameter defined expressed which edges are fairly big. We could find out the suitable value of this parameter by doing some trial and error. However, the value of this parameter is not influenced to the method efficiency but is influenced to consequence's accuracy. The parameter should respectively that, number of produced clusters be about the number of clusters in clustering metric.
- β parameter: This parameter is applied in clustering evaluation, (for F-Measure calculation). Setting this parameter with 5 is a good choice, in reality, we could assign other values to it, which is finally didn't impact on clustering efficiency. Notice that, in evaluating other methods based on this F-Measure, the parameter must be fixed.
- Number and length of documents, tag's length and kinds: these parameters influenced to clustering efficiency, especially, in parsing documents level means in tree structural presentation, so it expressed overall efficiency. User couldn't choose them but, could choose the set of documents.

Experimental Results

We have two kinds of data sets in evaluation, one available in [20], is English Single Label Categorization Collection-XML Document, which have 10 categories of documents, each category include 10 documents. And synthesize data set which is produce by piecing the set of tags with different depth and length but single subject   together in a single category; we need to use of 5 categories include 10 documents in each category.

The below figures presents the results of running the system, on both real and synthesizes data sets in different value of clustering factor and also single dimension.

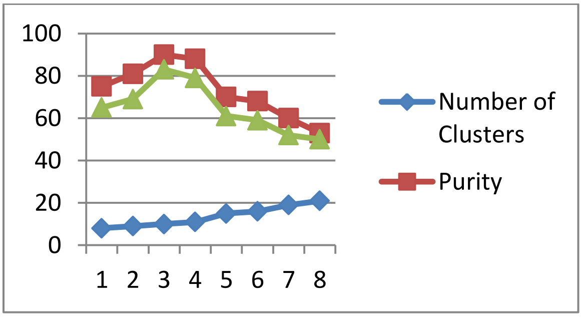


Figure 4: Diagram of running the system on real dataset in different value for Clustering Factor

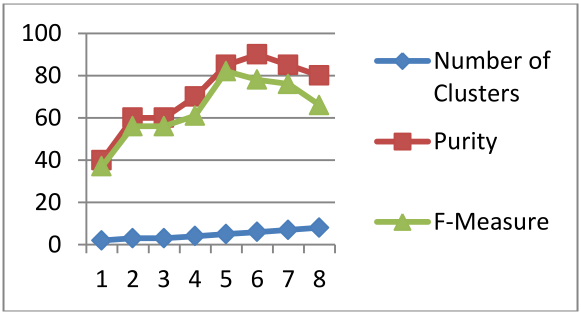


Figure 5: Diagram of running the system on synthesize dataset in different value for Clustering Factor

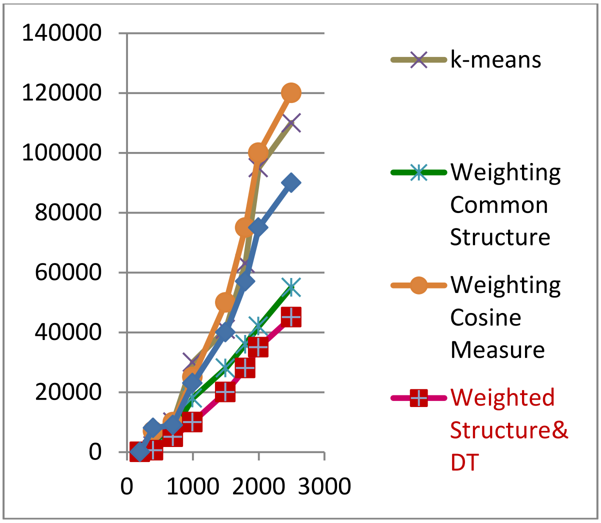


Figure 6: Execution Time Comparison on varying number of XML documents in msec

In this diagram, we compare common methods of clustering XML documents with proposed system on execution time, and we find out, our method due to using transferring to time series approach, getting the best information about the documents very quickly, so it can have less time to clustering documents too. In verse of other clustering methods, likes catching common structure in document's tree, which needed to search all the document.

Figure 7, 8 present result of running common and proposed system on different number of documents. Diagrams reveal proposed method do the best in a constant value of dimension and clustering factor parameter, as we expected.

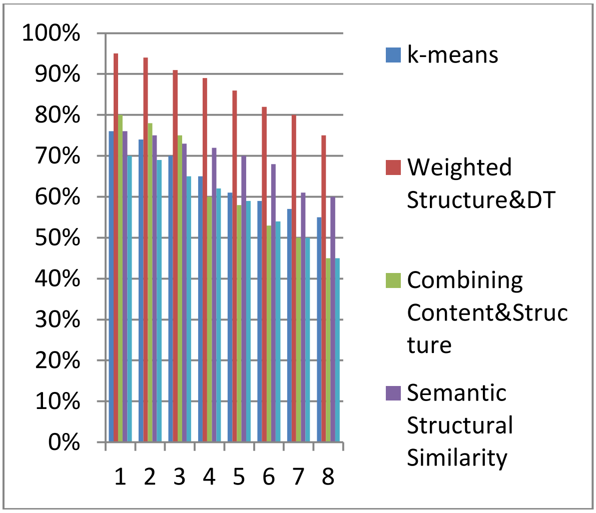


Figure 7: Purity of varying methods on different number of XML documents

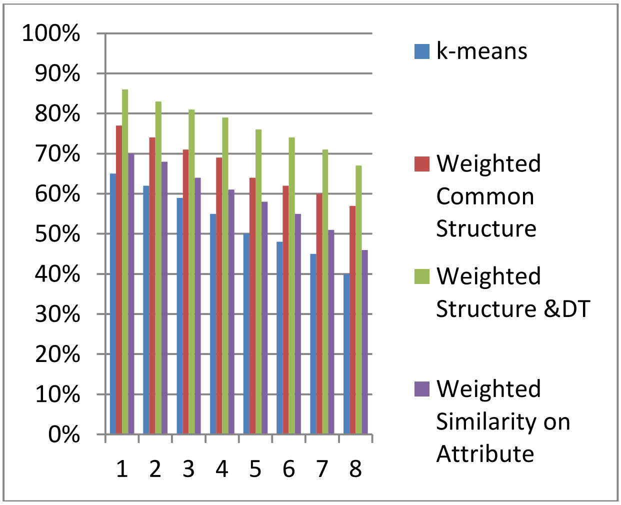


Figure 8: F-Measure of varying methods on different number of XML documents

Conclusions and Future works

Evaluation results are desirable enough. Clustering metric, clustering documents based on subject and content of documents. But our proposed method, cluster them based on structure of documents. So comparing this clustering method with clustering metric is suitable enough but not completely admirable. In fact, it should be a clustering metric based on documents structure to exploit in evaluating, or instead of using external metric, use internal metric. In order to evaluate clustering, we could analyze the distance of the cluster's intra points with each other and with others in the other clusters. This rate named internal metrics.

Efficiency of the proposed method depended on triangulation's efficiency which, in high dimension have lower performance or even impossible. In other hand, dimension of space, intensively effect on comparison accuracy. Consequently, we can proposed that make use of a simpler clustering method which, is suitable for lower number of points but, in high dimensions, can be a pretty instead. Mean that with applying another method instead of Delaunay, not only increasing value of sampling parameter but also get a desirable result, too.

References

1. Jeong Hee Hwang, Keun Ho Ryu. A weighted common structure based clustering technique for XML documents. Netherlands: Elsevier Inc. 2010.
2. Guo Yongming, Chen dehua, Le JIajin. Clustering XML Documents by Combining Content and Structure. International Symposium on Information Science and Engieering. 2008.
3. Lingxian Yang, Jinguang Gu, Heping Chen. Clustering Algorithm Based on Semantic Distance for XML Documents. First International Workshop on Database Technology and Applications. 2009.
4. Tae-Soon Kim, Ju-Hong Lee, Jae-Won Song. Semantic Structural Similarity for Clustering XML Documents. International Conference on Convergence and Hybrid Information Technology. 2008.
5. Alishahi M, Ravakhah M, Shakeriaski B, Naghibzade M. XML Document Clustering Based on Common Tag Names

Anywhere in the Structure. Proc. of the 14th International CSI Computer Conference. 2009.

6. Lei Liu, Yongqing Zheng, Baoshi Ding, Haiyan Liu. Methodology for clustering XML documents based on labeled tree. Sixth International Conference on Fuzzy Systems and Knowledge Discovery. 2009.
7. Jin-sha Yuan, Xin-ye Li, Li-na Ma. An Improved XML Document Clustering Using Path Feature. Fifth International Conference on Fuzzy Systems and Knowledge Discovery. 2008.
8. Naresh Kumar Nagwani, Ashok Bhansali. Clustering Homogeneous XML Documents Using Weighted Similarities on XML Attributes. IEEE co. 2010.
9. LI Wei, LI Xiong-fei, ZHAO Yan. XML Documents Clustering Research Based on Weighted Cosine Measure. Fifth International Conference on Frontier of Computer Science and Technology. 2010..
10. Abiteboul S, Buneman P, Suciu D. Data on the Web: From Relations to Semistructured Data and XML. 2000.
11. Nayak R. XML Data Mining: Process and Applications. Idea Group Inc. / IGI Global. 2008.
12. Nayak R. Fast and effective clustering of XML data using structural information. Knowl. Inf. Syst. 2008; 14: 197-215.
13. Flesca S, Manco G, Masciari E, Pontieri L, Pugliese A. Fast detection of XML structural similarities. IEEE Trans. Knowl. Data Engin. 2005; **7**: 160–175.
14. Denoyer L, Gallinari P. Report on the XML mining track at INEX 2005 and INEX 2006: categorization and clustering of XML documents. SIGIR Forum. 2007; 41: 79-90.
15. Dalamagas T, Cheng T, Winkel K, Sellis T. A methodology for clustering XML documents by structure. Inf. Syst. 2006; 31: 187-228.
16. Lian W, Cheung W, Mamoulis N, Yiu S. An Efficient and Scalable Algorithm for Clustering XML Documents by Structure. IEEE Trans. Knowl. Data Eng. 2004; 16: 82-96.
17. Lian W, Wai-lok D. An efficient and scalable algorithm for clustering XML documents by structure. IEEE Transaction on knowledge and Data engineering. 2004; 16: 82- 96.

18. Antonellis P, Makris C, Tsirakis N. XEdge: clustering homogeneous and heterogeneous XML documents using edge summaries. SAC. 2008; 1081-1088.
19. Alishahi M, Naghibzadeh M. Tag Name Structure-based Clustering of XML Documents. to be published in International Journal of Computer and Electrical Engineering (IJCEE). 2010.
20. http://wwwconnex.lip6.fr/~denoyer/wikipediaXML/
21. L. Denoyer P, Gallinari. The Wikipedia XML Corpus. 2006.
22. evaluation-of-clustering-1.html
23. http://www.cgal.org
24. Mundur P, Rao Y, Yesha Y. Key frame based video summarization using Delaunay clustering. International Journal on Digital Libraries. 2006; 6: 219–232.
25. http://www.ee.ucl.ac.uk/~mflanaga/java/