ISSN 2007-9737

Clustering XML Documents Using Structure and Content based on a New Similarity Function OverallSimSUX

Damny Magdaleno, Ivett E. Fuentes, and María M. García

Computer Science Department,
Universidad Central “Marta Abreu” de Las Villas (UCLV), Villa Clara,
Cuba

{dmg, ifuentes, mmgarcia}@uclv.edu.cu

Abstract. Every day more digital data in semi-structured format are available on the World Wide Web, corporate intranets, and other media. Knowledge management using information search and processing is essential in the field of academic writing. This task becomes increasingly complex and defiant, mainly because collections of documents are usually heterogeneous, big, diverse, and dynamic. To resolve these challenges it is essential to improve management of time necessary to process scientific information. In this paper, we propose a new method of automatic clustering of XML documents based on their content and structure, as well as on a new similarity function OverallSimSUX which facilitates capturing the degree of similarity among documents. Evaluation of our proposal by means of experiments with data sets showed better results than those in previous work.

Keywords. Clustering, XML, structure and content, similarity.

1 Introduction

An XML document is a hierarchical and auto-descriptive entity of information in a semi-structured format, since it incorporates structure and data in the same entity. Such structure of information can be used to retrieve relevant documents [1]. Being expandable, having a structure easy to analyze and process, XML has become the standard format of data exchange among Web applications [2]. Every day more digital data in XML format are available on the Web, corporate intranets, databases, and other media [2], so there is a need to manage these big volumes of data efficiently. Clustering based on XML structure and/or content [3] allows organizing the information.

A clustering algorithm tries to find natural clusters of data based mainly on similarity, so it is desirable that the objects that belong to the same cluster be as similar as possible and the objects that belong to different clusters be as dissimilar as possible [4].

In this paper, a new method for automatic clustering of XML documents is proposed using their content and structure. Another contribution is a new function of similarity OverallSimSUX which facilitates capturing the degree of similarity among the documents. The rest of the paper is organized as follows. Section 2 describes forms of clustering XML documents and some related papers; in Section 3 a new model of clustering of XML documents is presented using the new similarity function proposed by us. In Section 4 the experimental results are analyzed, and finally, Section 5 presents conclusions.

2 Related Work

Since XML documents are semi-structured, three forms of computing distance or similarity of these exist: (1) considering only the content of documents; (2) considering only the structure of documents; and (3) considering both dimensions of XML documents (structure and content).

The algorithms that consider only the content of documents obviate the advantage of structure which they offer as well. The algorithms that carry out lexical analysis, generally view a document as a bag of words, therefore, all the labels are

Only content	Kurgan, L. *Semantic mapping of xml tags using inductive machine learning* [7]	Use a variant of VSM.
	Shen, Y. *Clustering schemaless xml document* [8]	
Only structure	Dalamagas, T. *A Methodology for Clustering XML Documents by Structure* [2]	Use an XML tree representation to calculate a variant of tree-edit distance.
	Flesca, S. *Fast detection of XML structural similarities* [5]	
	Lesniewska, A. *Clustering XML documents by structure* [10]	
	Chawathe, S.S. *Comparing Hierarchical Data in External Memory* [11]	Consider XML structure based on the use of Edit Graph.
	Costa, G. *Hierarchical clustering of XML documents focused on structural components* [13]	Propose a new hierarchical approach.
	Aïtelhadj, A. *Using structural similarity for clustering XML documents* [12]	Follow the two-step approach to clustering XML documents.
Both structure and content	Kutty, S. *Combining the structure and content of XML documents for clustering using frequent subtrees* [16]	Use Closed Frequent Sub-Trees.
	Yang, W. *A semi-structured document model for text mining* [17]	Analyze a variant of XML document comparison based on VSM.
	Tekli, J.M. *A Novel XML Document Structure Comparison Framework based-on Subtree Commonalities and Label Semantics* [18]	Propose a framework to deal with both structural and semantic similarity in XML documents, use tree-edit distance.
	Pinto, D. BUAP: Performance of K-Star at the INEX'09 Clustering Task [19]	Use the iterative clustering algorithm K-Star in a recursive clustering process.

Fig. 1. Summary of XML clustering algorithms

eliminated and the structural information that the document offers is lost [5].

Following this focus, several authors based their research on the traditional Vector Space Model (VSM) [6] representation [7, 8].

Several works treat an XML document as a tree taking advantage of its hierarchical structure. Examples of this approach are [2, 9, 10] which use a tree representation to calculate the tree-edit distance or some of its variants to compare the documents. The method of Structural Summaries is proposed in [2] to reduce nesting and repetitions which may exist in the trees. Other methods of documents' clustering considering their structure are based on the use of Edit Graph [11]. In [12], a new hierarchical approach is proposed which allows considering multiple forms of structural components to structurally isolate homogeneous clusters of XML documents.

At each level of the resulting hierarchy, clusters are divided by considering some type of structural components which still differentiate structures of XML documents. In [13] SOS is proposed, a similarity search method based on structures and styles of office documents. SOS needs to compute similarity values between multiple pairs of XML files included in the office documents. The authors also proposed LAX+, which is an algorithm to calculate a similarity value for a pair of XML files by matching leaf nodes of sub-trees in the XML files. A new method for clustering XML documents is proposed in [14], where the goal is to group documents sharing similar structures, following a two-step approach. Firstly they extract automatically the structure from each XML document to be classified. The extracted structure is then used as a representation model to classify the corresponding XML document. The idea behind clustering is that if XML documents share similar structures, they are more likely to correspond to the structural part of the same query.

Most of state of the art research does not use the two dimensions (structure and content) because of their great complexity [15]. However, to obtain better results in clustering it is essential to use both [16]. A first and very simple option is to mix the content and the labels of a document in a VSM. In [16] the Closed Frequent Sub-Trees method is used to process the structure of documents and then to perform preprocessing of the content of the documents.

Other works developed extensions to the VSM representation called C-VSM and SLVM [17]. However, C-VSM can be seen as a method of "low contribution" since it ignores the semantic relationships among different elements; and SLVM does not consider the relationships among common elements, so it can be seen as a technique of "over contribution". With the purpose of resolving these problematic issues, in [5] the Proportional Transportation Similarity is proposed, which works with heavy comparisons according to likeness or unlikeness of the elements while comparing pairs of documents. In [18] a framework is suggested to deal with both structural and semantic similarities in XML documents. This framework consists of four main modules for discovering structural commonalities among sub-trees, identifying sub-tree semantic resemblances, computing tree-based edit operation costs, and computing tree- edit distance. In [19] unsupervised classification techniques are used in order to group documents of a given huge collection into clusters. The authors approached this challenge by using the iterative clustering algorithm K-Star [20] in a recursive clustering process over sub-sets of the complete collection. Fig. 1 presents a summary of previous XML clustering algorithms.

3 Clustering Model

This section presents a method of automatic clustering of XML documents, as well as a new similarity function *OverallSimSUX* which facilitates capturing the degree of similarity among the documents.

3.1. OverallSimSUX Similarity Matrix

Meditating shortly on the concept of a document, there can be found multiple types of documents, so it seems more natural to treat them as a set of parts (i.e. scientific papers, news, etc.). Consequently given a document *d*, a set of structural units $SU = \{SU_1,\ldots, SU_n\}$ can be associated with it. For example, in a scientific paper, structural units will be abstract, introduction, Section 1, etc.

The existent structural relationships among XML documents can contribute to better clustering results when the content is used in function of the relations between their *SU*. In this paper, for the construction of a similarity matrix, a new measure of similarity is proposed which facilitates capturing the degree of similarity of these documents. In this function the existent relationship among the documents is analyzed, treating *SUs* simultaneously like independent collections and the documents like indivisible units.

Fig. 2 shows how a similarity matrix *OverallSimSUX* is obtained starting from a collection of XML documents. For the matrix construction, it is necessary to perform three steps: (1) to build a first representation, denominated *Representation I*, using the *SU* of the documents; (2) to build a second representation, *Representation II*, by considering the whole collection; (3) to carry out clustering using *Representation I*.

a. Representation I (Step 1)

The original collection of documents is divided in *n* collections, were *n* is the number of SUs in a document. Definition 1 captures the correspondence between the collection and *SU*, giving place to the *k*-collection concept.

Definition 1 (*k*-collection). Let *D* be a corpus of XML documents, then the *k*-collection of the collection *D* is formed by the group of new documents DSU_k:

$$DSU_k = \{SU_k \in d, \forall d \in D\}, \tag{1}$$

where *d* is a document of *D*, Su_k is the *k*-th *SU* of *d*.

For each *k*-collection, *Representation I* is built using the classic VSM. In particular, the construction of this matrix was carried out by means of the Term Frequency and Inverse Document Frequency (*TF-IDF*) measure [6]. *TF-IDF* is a statistical measure of weight often used in natural language processing to determine

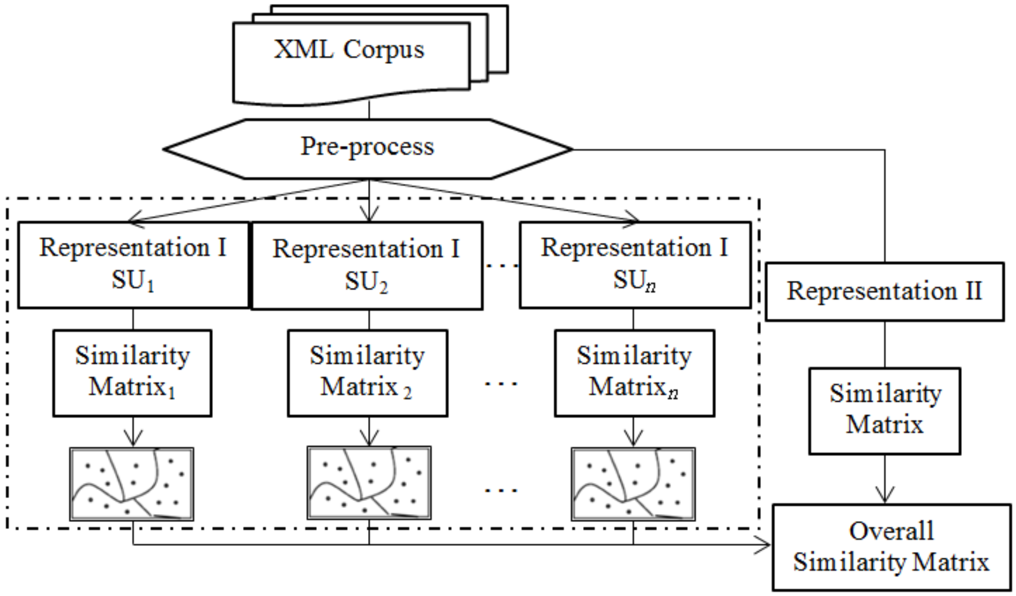

Fig. 2. This diagram shows the methodology to build the OverallSimSUX similarity matrix

how important a term is in a given corpus, by using a vector representation. The importance of each term increases proportionally to the number of times this term appears in the document (frequency), but is offset by the frequency of the term in the corpus.

The *tf* component of the formula is calculated by the normalized frequency of the term, whereas *idf* is obtained by dividing the number of documents in the corpus by the number of documents which contain the term, and then taking the logarithm of that quotient. Given a corpus DSU_k and a document d_j ($d_j \in DSU_k$), the *TF-IDF* value for a term t_i in d_j is obtained by the product between the normalized frequency of the term t_i in the document d_j (tf_{ij}, equation 2) and the inverse document frequency of the term in the corpus ($idf(i)$, equation 3) as follows [19]:

$$tf_{ij} = \frac{frecuency(t_i,d_j)}{\sum_{s=1}^{|d_j|} frecuency(t_s,d_j)}, \quad (2)$$

$$idf(i) = log\left(\frac{|D|}{|d:t_i \in d, d \in D|}\right), \quad (3)$$

$$tf_{ij}\text{-}idf_i = tf_{ij} \times idf(i). \quad (4)$$

b. Representation II (Step 2)

In this paper the structure of the documents is added to the analysis, therefore, in *Representation II* a modification to the classic VSM is carried out and the frequency will be weighted by the *SU* that corresponds to the analyzed term. This approach was proposed in [21]. Equation 5 shows how to calculate this frequency in a document *j* for a term *i*:

$$tf_{ij} = \sum_{k=1}^{n} (w_{kj} \times frecuency_{ik}), \quad (5)$$

$$w_{kj} = \left(e^{\left(-L_{SU}/L_{Doc}\right)}\right)^{pot}, \quad (6)$$

where n is the quantity of the *SU* present in document *j*, $frecuency_{ik}$ is the frequency of term *i* in SU_k, w_{kj} is the weight calculated for SU_k in the document *j*, L_{SU} is the length of the SU_k, L_{Doc} is the length of the document *j*, *Pot* is a given value. After several experiments, the best results were obtained with a *Pot* value of 5.

c. *k*-collection Clustering (Step 3)

Starting from *Representation I*, a similarity matrix is calculated, which compares two documents using the Cosine measure; this is shown in equation 7. For each k-collection an independent cluster is obtained.

$$S_{Cosine}(d_i, d_j) = \frac{\sum_{r=1}^{s}(d_{ir} \times d_{jr})}{\sqrt{\sum_{r=1}^{s} d_{ir}^{2} \times \sum_{r=1}^{s} d_{jr}^{2}}}. \quad (7)$$

To carry out clustering, the classic *K*-Star algorithm [20] is used. Nevertheless, the choice of other algorithm does not invalidate the idea of the method proposed in this work. In future work, we will present a comparative study of the performance of our approach against other clustering techniques. *K*-Star is an iterative clustering method that starts by building a similarity matrix of the documents to be clustered (corpus). This algorithm does not need to know the number of cluster value a priori, instead it automatically proposes a number of clusters in a totally unsupervised way. *K*-Star is a considerably fast algorithm and also obtains reasonably good results when applied to text corpora [22].

d. OverallSimSUX Matrix Calculation

The considerations exposed before are the starting point to develop the similarity measure *OverallSimSUX*, specified formally by Definition 3. It begins with the results of the clustering carried out for all k-collections and the similarity matrix based on the calculation of the cosine measure using *Representation II*.

Definition 2 (λ-membership). This is a boolean function, i.e. one if both documents (i, j) belong to the same cluster c_n, otherwise it is zero depending on clustering results by using *Representation I*. The λ-membership is formalized in equation 8:

$$\lambda(i,j) = \begin{cases} 1, & \{i,j\} \in c_n \\ 0, & i \in c_n \wedge j \in c_m \mid m \neq n \end{cases}. \quad (8)$$

Definition 3 (OverallSimSUX). A normalized measure of similarity among the documents i, j is considered. It is calculated by the function $S_{OSSUX}(i,j)$, equation (9), and its values are within [0,1]:

$$S_{OSSUX}(i,j) = \frac{\sum_{k=1}^{n}(w_k \times \lambda_k(i,j)) + S_g(i,j)}{\sum_{k=1}^{n}(w_k) + 1}, \quad (9)$$

```
1.Build the OverallSimSUX similarity
  function.
2.Estimate similarity thresholds.
3.Apply K-Star clustering method from
  OverallSimSUX matrix.
```

Fig. 3. Basic steps to obtain document clustering using K-Star method with OverallSimSUX matrix

```
Input: Corpus D of XML documents
Output: Set of clusters, cluster
quality, the most representative
document of each cluster.
Begin
1. Pre-process /* lexical analysis, stop
   word elimination, stemming… */
2. Build all k-collections(corpus D)
3. For each DSUk
   - Rep-I←Make Representation-I(DSUk)
           according to TF-IDF
   - Sim_matrix←Calculate similarity
                matrix for Rep-I using
                Cosine measure
   - Clusters←Apply K-Star clustering
              method to Sim_matrix
  end_for
4. Rep-II←Make Representation-II of
         entire corpus D using equation
        (5) for calculating frequency
5. Sim_matrixII←Calculate similarity
                matrix for Rep-II using
                Cosine measure
6. O_Sim_Matrix←Calculate similarity
                Matrix using OverallSimSUX
                measure taking into
                account all the clustering
                of all DSUK and
                Sim_matrixII
7. Make final clustering applying K-Star
   clustering method to O_Sim_matrix
end
```

Fig. 4. General procedure

where $S_g(i,j)$ is an element of the matrix S_g calculated by equation (7) from *Representation II*; w_k is the weight of SU_k, n is the quantity of *SUs* identified in the documents, $\lambda_k(i,j)$ is λ-membership of the documents i, j from the clustering result for *Representation I* of SU_k.

ISSN 2007-9737

3.2. Final Clustering

To carry out final clustering, the *K*-Star algorithm is used again.

The K-Star algorithm uses a similarity threshold for determining the minimum ratio of similarity that must exist between a document and an already formed cluster in deciding whether to incorporate a given document as a member of this cluster. This threshold must be given [19] or calculated using several variants [23]. In this work we calculated the threshold using the means of all similarity values between all pairs of documents.

Fig. 3 shows the basics steps to obtain the final clustering. A complete description of the implemented approach is given in the following section.

3.3 Description of the Approach

Fig. 4 shows the proposed general procedure.

This procedure includes three important steps: (1) preprocessing of the entire collection, identifying each Structural Unit; (2) textual representation using *Representation I* and *Representation II*, and (3) final clustering process.

4. Experimental Results

To check the validity of the obtained results starting from the pattern of the proposed clustering, two experiments were designed and applied to three data sets with the purpose of carrying out a statistical analysis. This analysis allows verifying if significant differences exist between the proposed methodology and other variants of algorithms reported in the literature.

4.1 Case Studies and Experiment Design

- *Case study 1:* documents retrieved from the site of ICT of the Center of Studies on Informatics of the Universidad Central "Marta Abreu" de las Villas (UCLV)[1].

[1] http://ict.cei.uclv.edu.cu

- *Case study 2:* summary of documents of the IDE-Alliance repository, provided by the University of Granada, Spain.
- *Case study 3:* summary of Wikipedia documents, published by INEX to evaluate clustering.

We conform 16 corpora of XML documents (corpora 1 to 7 with documents from case study 1; corpora 8 to 11 with documents from case study 2, the rest from case study 3).

We perform two experiments with these 16 corpora for evaluating the results according to our objective.

The first experiment consisted in verifying how our method behaves globally on the three data sets described previously. Other methods used for comparison were the one proposed in INEX [19] broadly used for clustering of XML documents and the *K*-Star algorithm, a predecessor of our proposal. Both approaches, the K-Star algorithm and the algorithm proposed in [19], were implemented in a system for clustering scientific papers in XML format (LucXML).

To evaluate the results, we applied an external measure called *Overall F-measure* [24]. This measure is based on *Precision* (*Pr*) and *Recall* (*Re*) [25]. *Pr* and *Re* are calculated for a cluster *g* and a class *c* as follows:

$$Pr(c,g) = \frac{n_{cg}}{n_g}, \tag{10}$$

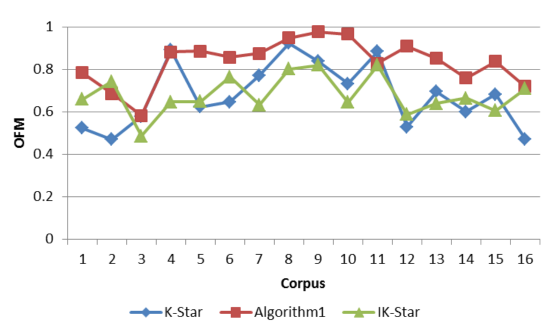

Fig. 5. Overall *F*-measure values of the compared algorithms

Table 1. Values of the Micro-Purity and Macro-Purity measures for INEXK-Star and the Alg1 (our approach)

Corpus	Micro-Purity		Macro-Purity	
	IK-Star	Alg1	IK-Star	Alg1
1.	**0.716**	0.568	**0.792**	0.534
2.	**0.559**	0.451	**0.488**	0.483
3.	0.3	**0.673**	0.2	**0.759**
4.	0.405	**0.418**	0.405	**0.628**
5.	**0.725**	0.418	**0.816**	0.628
6.	0.459	**0.557**	0.44	**0.643**
7.	0.481	**0.571**	0.481	**0.596**
8.	0.511	**1**	0.511	**1**
9.	0.542	**0.545**	0.656	**0.697**
10.	0.636	**1**	0.627	**1**
11.	0.463	**0.633**	0.529	**0.71**
12.	0.722	**0.958**	0.792	**0.972**
13.	0.59	**0.597**	0.59	**0.664**
14.	0.552	**0.689**	0.552	**0.806**
15.	0.525	**0.638**	0.524	**0.71**
16.	**0.515**	0.461	**0.56**	0.55

$$Re(c,g) = \frac{n_{cg}}{n_c}, \quad (11)$$

where n_{cg} is the number of objects of class c in cluster g, n_g is the number of objects in cluster g, and n_c is the number of objects of class c. These values are used to calculate *F*-measure using harmonic means of *Pr* and *Re* as shown in formula 12:

$$FM(c,g) = \frac{1}{\alpha\left(\frac{1}{Pr(c,g)}\right)(1-\alpha)\left(\frac{1}{Re(c,g)}\right)}. \quad (12)$$

If α=1, then $FM(c,g)$ coincides with *Pr* value; if α=0, $FM(c,g)$ coincides with *Re* value. So α=0.5 means that *Pr* and *Re* have equal weight. Finally, the Overall *F*-measure is calculated using expression 13:

$$OFM = \sum_{i=1}^{k} \frac{n_c}{n} max\{FM(c,g)\}. \quad (13)$$

Results obtained for the Overall *F*-measure are shown in Fig. 5 where the method of [19] is denoted as IK-Star and our approach is denoted as Algorithm1.

In the second experiment it was verified how our proposal behaves globally on 16 corpora of the three data sets described previously. Here we compared our method with the method of INEX'09 [19] using a technique based on the *Micro-Purity* and *Macro-Purity* measures described in [26]. This will show the quality of the groups obtained in each clustering. *Purity* is measured as the ratio of the number of documents with the majority label in a given cluster to the number of documents in this cluster. *Macro-* and *Micro- Purity* of the entire clustering solution is obtained as a weighted sum of the individual cluster purity. In general, the larger the value of purity, the better the clustering solution is. Table 1 presents these values.

Equations 15 and 16 show expressions for *Micro-Purity* and *Macro-Purity*, respectively:

ISSN 2007-9737

Table 2. Wilcoxon test statistics of results for OFM of Alg1 and OFM of K-Star

Experiment1		N	Mean Rank	Sum of Ranks	Alg1 - *K*-Star	
ofm_Alg1 - ofm_K-Star	Negative Ranks	2[a]	3.00	6.00	Z	-3.206[a]
	Positive Ranks	**14**[b]	9.29	130.00	Aymp. Sig (2-tailed)	**0.001**
	Ties	0[c]				
	Total	16				

(a. Alg1 < *K*-Star b. Alg1> *K*-Star c. Alg1= *K*-Star a. Based on positive ranks)

Table 3. Wilcoxon test statistics of results for OFM of IK-Star and OFM of Alg1

Experiment1		N	Mean Rank	Sum of Ranks	IK-Star - Alg1	
ofm_IK-Star - ofm_Alg1	Negative Ranks	**15**[a]	8.87	133.00	Z	-3.361[a]
	Positive Ranks	1[b]	3.00	3.00	Aymp. Sig (2-tailed)	**0.001**
	Ties	0[c]				
	Total	16				

(a. IK-Star< Alg1 b. IK-Star > Alg1 c. Alg1= IK-Star a. Based on positive ranks)

Table 4. Wilcoxon test statistics of results for MicroPurity of IK-Star and MicroPurity of Alg1

Experiment2		N	Mean Rank	Sum of Ranks	IK-Star - Alg1	
microP_IK-Star - microP_Alg1	Negative Ranks	**12**[a]	8.96	107.50	Z	-2.043[a]
	Positive Ranks	4[b]	7.13	28.50	Aymp. Sig (2-tailed)	**0.04**
	Ties	0[c]				
	Total	16				

(a. IK-Star< Alg1 b. IK-Star > Alg1 c. Alg1= IK-Star a. Based on positive ranks)

$$Purity(q) = \frac{NDMLC_q}{NDC_q}, \quad (14)$$

$$Micro-Purity(q) = \frac{\sum_{q=0}^{n} Purity(q) \times TFC(q)}{\sum_{q=0}^{n} TFC(q)}, \quad (15)$$

$$Macro-Purity(q) = \frac{\sum_{q=0}^{n} Purity(q)}{TotalofCategories}, \quad (16)$$

where *NDMLC* is the number of documents with the majority label in a cluster, *NDC* is the number of documents in the cluster, *TFC* is the total number of documents found by class, *TotalofCategories* is the number of clusters found by the clustering algorithm.

In general, best results were achieved by Algorithm 1 (our proposal) in both experiments.

Table 5. Wilcoxon test statistics of results for MacroPurity of IK-Star and MacroPurity of Alg1

Experiment2		**N**	**Mean Rank**	**Sum of Ranks**		**IK-Star - Alg1**
macroP_IK-Star - macroP_Alg1	Negative Ranks	**12**[a]	9.25	111.00	Z	-2.223[a]
	Positive Ranks	4[b]	6.25	25.00	Aymp. Sig (2-tailed)	**0.026**
	Ties	0[c]				
	Total	16				

(a. IK-Star< Alg1 b. IK-Star > Alg1 c. Alg1= IK-Star a. Based on positive ranks)

In order to validate the obtained results, for the first experiment we computed the Friedman test (Friedman two-way analysis of variances by ranks) [27, 28]. This test is a multiple comparison procedure capable to detect significant differences between the behaviors of two or more algorithms; i.e., it can be used for discovering whether at least two of the samples represent populations with different median values or not, in a set of n samples ($n \geq 2$).

Using a significance level of 0.05, corresponding to the 95% confidence interval, the Friedman test suggests rejecting the null hypothesis (p-value<0.05), therefore, there exist highly significant differences between at least two algorithms. Algorithm 1 are best ranked; however, this information cannot be used to conclude that our proposal is involved in these differences; for this reason we applied the Wilcoxon signed ranks test [29]. The latter is a pairwise procedure that aims to detect significant differences between two sample means, that is, the behavior of two algorithms. Using a significance level of 0.05, corresponding to the 99% confidence interval, the Wilcoxon test suggests to reject the null hypothesis (p-value<0.05) for the following pairwise comparisons: Alg1 vs. K-Star (0.001), Alg1 vs. IK-Star (0.001). These results statistically confirm that the proposed algorithm (Alg1) obtained better result than the other algorithms used in these case studies.

The non-parametric Wilcoxon test was also applied for the values of Purity measure calculated from all clustering obtained using our approach and the method proposed in [19]. Using a significance level of 0.05, corresponding to the 99% confidence interval, the results are 0.04 for the Micro-Purity results and 0.026 for the Macro-Purity results. In Tables 2-5 the values of significance of these tests are given.

5 Conclusions

The new proposed method of clustering of XML documents by using both content and structure allowed obtaining results similar or higher than the results of other methods used for comparison in three case studies with two experiments.

The new methodology for calculating the similarity function OverallSimSUX facilitates capturing of the similarity degree among the documents.

In future, we will study the effects of other clustering techniques (for example, fuzzy and hierarchical techniques) on the results of the proposed methodology.

References

1. **Guerrini, G., Mesiti, M., & Sanz, I. (2006).** An Overview of Similarity Measures for Clustering XML Documents. *Information Systems,* Athena Vakali and George Pallis (eds.).
2. **Dalamagas, T., Cheng, T., Winkel, K.J., & Sellis, T. (2006).** A Methodology for Clustering XML Documents by Structure. *Information Systems,* Athena Vakali and George Pallis (eds.).
3. **Tien, T. (2007).** Evaluating the Performance of XML Document Clustering by Structure only. *5th*

International Workshop of the Initiative for the Evaluation of XML Retrieval.

4. **Kruse, R., Döring, C., & Lesor, M.J. (2007).** Fundamentals of Fuzzy Clustering. *Advances in Fuzzy Clustering and its Applications*, J.V.d. Oliveira and W. Pedrycz (Eds.), John Wiley and Sons, England, pp. 3–27.
5. **Wan, X. & Yang, J. (2006).** *Using Proportional Transportation Similarity with Learned Element Semantics for XML Document Clustering*. International World Wide Web Conference Committee.
6. **Salton, G., Wong, A., & Yang, C.S. (1975).** A vector space model for automatic text retrieval. *Communications of the ACM,* Vol. 18, No. 11, pp. 613–620.
7. **Kurgan, L., Swiercz, W., & Cios, K.J. (2002).** Semantic mapping of xml tags using inductive machine learning. *11th International Conference on Information and Knowledge Management,* Virginia, USA.
8. **Shen, Y. & Wang, B. (2003).** Clustering schemaless xml document. *11th international conference on Cooperative Information System*.
9. **Flesca, S., Manco, G., Masciari, E., Pontieri, L., & Pugliese, A. (2005).** Fast detection of XML structural similarities. *IEEE Trans. Knowl. Data Engin*., Vol. 7, No. 2, pp. 160–175. DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/TKDE.2005.27
10. **Lesniewska, A. (2009).** Clustering XML documents by structure. *ADBIS'09 Proceedings of the 13th East European conference on Advances in Databases and Information Systems*.
11. **Chawathe, S.S. (1999).** Comparing Hierarchical Data in External Memory. *Proceedings of International Conference on Very Large Databases*.
12. **Costa, G., Manco, G., Ortale, R., & Ritacco, E. (2013).** Hierarchical clustering of XML documents focused on structural components. *Data & Knowledge Engineering*, Vol. 84, pp. 26–46 doi:10.1016/j.datak.2012.12.002
13. **Yousuke, W., Hidetaka, K., & Haruo, Y. (2013).** Similarity search for office XML documents based on style and structure data. *International Journal of Web Information Systems,* Vol. 9 No. 2, pp. 100–117. doi:10.1145/2428736.2428761
14. **Aïtelhadj, A., Boughanem, M., Mezghiche, M., & Souam, F. (2012).** Using structural similarity for clustering XML documents. *Knowledge and Information Systems,* Vol. 32, No. 1, pp. 109–139. doi: 10.1007/s10115-011-0421-5
15. **Tran, T., Nayak, R., & Bruza, P. (2008).** Combining Structure and Content Similarities for XML Document Clustering. *Seventh Australasian Data Mining Conference,* Glenelg, Australia.
16. **Kutty, S., Tran, T., Nayak, R., & Li, Y. (2008).** Combining the structure and content of XML documents for clustering using frequent subtrees. *INEX*, pp. 391–401.
17. **Yang, W. & Chen, X.O. (2002).** A semi-structured document model for text mining. *Journal of Computer Science and Technology*, Vol. 17, No. 5, pp. 603–610.
18. **Tekli, J.M. & Chbeir, R. (2011).** *A Novel XML Document Structure Comparison Framework based-on Subtree Commonalities and Label Semantics*. Vol. 11, Elsevier.
19. **Pinto, D., Tovar, M., & Vilariño, D. (2009).** BUAP: Performance of K-Star at the INEX'09 Clustering Task. *INEX 2009 Workshop Pre-proceedings,* Woodlands of Marburg, Ipswich, Queensland, Australia.
20. **Shin, K. & Han, S.Y. (2003).** Fast clustering algorithm for information organization. *Proc. of the CICLing Conference*, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 619–622.
21. **Magdaleno, D., Fernández, J., Huete, J., Arco, L., Fuentes, I., Artiles, M., & Bello, R. (2011).** New Textual Representation using Structure and Contents. *Research in Computing Science,* Vol. 54, pp. 117–130.
22. **Perez-Tellez, F., Pinto, D., Cardiff, J., & Rosso, P. (2009).** Improving the Clustering of Blogosphere with a Self-term Enriching Technique. *Text, Speech and Dialogue*, Springer, pp. 40–47.
23. **Ruiz-Shulcloper, J. (1995).** *Introducción al reconocimiento de patrones. Enfoque lógico combinatorio*. México, CINVESTAV IPN.
24. **Steinbach, M., Karypis, G., & Kumar, V. (2000).** A comparison of document clustering techniques. *Proceedings of 6th ACM SIGKDD World Text Mining Conference*, Boston, ACM Press.
25. **Frakes, W.B. & Baeza-Yates, R. (1992).** *Information Retrieval. Data Structure & Algorithms*. New York, Prentice Hall.
26. **Vries, C., Nayak, R., Kutty, S., Geva, S., & Tagarelli, A. (2011).** Overview of the INEX 2010 XML mining track: clustering and classification of XML documents. *Lecture Notes in Computer Science*, Vol. 6932, Springer, pp 363–376.
27. **Friedman, M. (1937).** The use of ranks to avoid the assumption of normality implicit in the analysis of

variance. *Journal of the American Statistical Association*, Vol. 32, No. 200, pp. 675–701.

28. **Friedman, M. (1940).** A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, Vol. 11, No. 1, pp. 86–92.
29. **Wilcoxon, F. (1945).** Individual comparisons by ranking methods. *Biometrics Bulletin*, Vol. 1 No. 6, pp. 80–83.

Damny Magdaleno received his B.Sc. degree in Computer Science from the Universidad Central "Marta Abreu" de Las Villas (UCLV), Santa Clara, Cuba, in 2006, his M.Sc. degree in Computer Science from UCLV in 2008. He is with the Artificial Intelligence Lab of the Center of Studies on Informatics, UCLV, where he currently works toward the Ph.D. degree. He has published monographs and papers in national and international conferences and journals. His research interests include soft computing, text mining, and machine learning.

Ivett E. Fuentes received her B.Sc. degree in Computer Science from the Universidad Central "Marta Abreu" de Las Villas (UCLV), Santa Clara, Cuba, in 2013. She is with the Artificial Intelligence Lab of the Center of Studies on Informatics, UCLV, where she currently works toward the M.Sc. degree. She has published papers in national and international conferences and journals. Her research interests include soft computing, text mining, and machine learning.

Maria M. García received her B.Sc. degree in Mathematical Cybernetics from the Universidad Central "Marta Abreu" de Las Villas (UCLV), Santa Clara, Cuba, in 1985 and her Ph.D. in Technical Sciences from UCLV in 1997. She is a full professor at the Artificial Intelligence Lab of the Center of Studies on Informatics, UCLV, where she is also the General Coordinator of the Master in Computer Science Program (AUIP awarded). Also, she exhibits a long record of academic exchange with many universities in Latin America and Europe. Dr. Garcia has authored 6 books, published over 150 papers in conference proceedings and scientific journals, and supervised several Bachelor, Master and Ph.D. theses. She received numerous prestigious awards from the Cuban Academy of Sciences and other renowned scientific societies. Her research interests include neural networks, soft computing, machine learning, and pattern recognition.

Article received on 13/12/2013, accepted on 14/10/2014. Corresponding author is Damny Magdaleno.