New Textual Representation using Structure and Contents

Damny Magdaleno[1], Juan M. Fernández[2], Juan Huete[2], Leticia Arco[1], Ivett E. Fuentes[1], Michel Artiles[1], and Rafael Bello[1]

[1] Computer Science Department, Central University "Marta Abreu" from Las Villas, Camajuaní Road km 5½, Santa Clara, Villa Clara, Cuba
[2] Computer Science and Artificial Intelligence Department University of Granada, Granada, Spain
{dmg, leticiaa, ifuentes, mae}@uclv.edu.cu
{jmfluna, jhg}@decsai.ugr.es

Abstract. The effectiveness of documents representation is directly related with how well can be compared their contents with another. When representing XML documents it is important not only its content, the structure can be exploited in tasks of text mining. Unfortunately, most XML documents representations do not consider both components. In this paper is presented a new form of textual XML documents representation using their structure and contents. The main results are: the new form of textual representation, following the criterion that depending on the location in which is presented a term within a document will have more or less importance in deciding how relevant this is in the document; it was joined to GARLucene software, increasing its potential for handling XML documents; the clustering, based on differential Betweenness of 25 textual collections represented with the new proposal, yielded better results than when they were represented with classic VSM.

Keywords: Textual representation, XML, clustering and document management.

1 Introduction

The increase of information in digital format, facilitated by storage technologies, poses new challenges to information processing tasks, among which may include: information retrieval, clustering and classification [1].

In performing these tasks, one of the steps is the Textual Representation (TR), which aims to transform textual document into a format that is suitable for input to algorithms application (e.g. machine learning, clustering and classification) in order to do Text Mining (TM) [2].

The effectiveness of a document representation is directly related to the accuracy with which the selected set of terms represents the document's contents and how well can be

compared that document's contents with another, that is, given two documents d1 and d2 and its representations r1 and r2, respectively, if r1 equals r2, this means that the content of d1 is equal to the contents of d2 with a level of abstraction [1]. So the TR have a key role in manipulating text documents, then a textual representation leads to good results in tasks such as clustering.

Among the different techniques of TR in the literature may be mentioned the Vector Space Model (VSM) [3], which is widely recognized as an effective representation for documents in the TM community, especially in the areas of information retrieval, clustering and classification. This representation sees the documents as a set of vectors where each dimension represents the weight of a term in the contents, which can be calculated, rather easily based on the number of term occurrences in the document, for example using inverse document frequency, or if exits information on the categories of documents using the Shannon entropy on all documents class set, for which is used the classification information [4]. In [5] the representation used is based on phrases rather than words to form the vector representation, using such phrases as input units for the functions of traditional weighting: Binary, TF and TF-IDF. In [6] are proposed the CONSOM model using two vectors instead of one to represent both the input documents with the aim of combining the vector space with what they call a conceptual space. The use of self-organizing maps with fuzzy logic for the RT content of Web pages was proposed in [4].

Some authors state that the documents are indivisible and independent units. Reflecting briefly on the concept of a document, can be found multiple types where it is more natural to treat them as a set of parts, these include scientific papers, which usually consist of title, abstract, keywords, a series of sections (can be divided into several subsections and so on), conclusions, among others. Therefore, given a set of documents $D = \{d_1, \ldots, d_m\}$, these correspond to a set of structural units $U = \{u_1, ..., u_n\}$. In this way the concept of document as indivisible unit disappears. Such is the case of XML format documents (Extensible Markup Language), which is a meta-language developed by the W3C[1] that arose from the need that the company had to store large amounts of information. An XML document is a self-descriptive hierarchical structure of information, which consists of a set of atoms, compound elements and attributes [7]. Added to this XML documents contain information on a semi-structured form [8], incorporating data and structure in the same entity. XML are extensible, with easy analysis and processing structure. The labels in XML documents allowing semantic content description of the elements. Thus, the structure of documents can be exploited for retrieval of relevant documents [9]. For all the above, XML documents are undoubtedly the standard data exchange format between Web applications and everyday more electronic data are presented on the web in this format [7]. For efficient organization and retrieval of relevant documents, a possible solution is to clustering XML documents based on their structure and / or its content [10]. A clustering algorithm attempts to find natural clusters of data based mainly on the similarity and relationships of objects, so as to obtain the internal distribution of the data set by its

[1] http://www.w3c.org.

partitioning into clusters. When the clustering is based on the similarity of the objects, it is intended that objects belonging to the same cluster are as similar as possible and the objects belonging to different clusters are as different as possible [11].

Therefore a proposal for TR for XML documents is presented in this paper, using the existing structure and contents therein, specifically dealing with the content in terms of the document's structure, following the criterion that depending on the location (Structural Unit, SU) in that a term (word) is present inside a document, it will have more or less importance in deciding how relevant this is in the paper. This representation will be used in a document clustering algorithm. The clustering algorithm applied in this paper is based on Differential Betweenness (DB) [12], which has shown good performance in textual domains. The organization of the paper is as follows: Section 2 shall treat forms of XML documents representation and related work; section 3 presents a new form of representation that weighs textual content based on the structure; in 4, application of the technique implemented in a system for managing documents is shown; section 5 will discuss the experimental results and finally; Section 6 presents conclusions.

2 XML Document Representation Forms

When the XML semi-structured documents, there are three ways to make textual representation: (1) representation that considers only the contents of the documents, (2) representation which considers only the structure, and (3) representation that considers these two dimensions of XML documents (structure and content).

2.1 Only Content Representation

This type of representation is often presented in the literature, but in the case of XML documents it ignores the advantage they offer, its structure. Thus, this approach focuses on treating the documents only by their content, either by performing a lexical analysis only, or including syntactic or semantic elements in the study. Those algorithms that perform lexical analysis, generally considered the documents as a bag of words, therefore, removed all the labels and lose the structural information provided by the documents [13]. Following this approach several authors rely on the traditional VSM representation.

2.2 Only Structure Representation

Making a TR of XML documents considering only its hierarchical structure is vitally important in tasks such as clustering, information extraction and integration of heterogeneous data, among others [9]. Several works represent XML documents in tree using its hierarchical structure, an example of this is made by [7, 14] using the tree view

to calculate the tree-edit distance or some variant to compare documents, this is just the number of operations (insertion, remove and replacement of nodes) to perform in a tree so that its structure is equal to the other tree with which it is compared. The smaller the number of operations is, the greater the similarity between the trees for XML documents. In [7] is proposed the Structural Summaries calculation for reducing trees to compare, given their nests and repetitions that may exist. Thus, representations are obtained as low as possible to maintain the relationships between elements of the tree and facilitate later comparisons. Other forms of document representation considering structure are based on the use of Edit Graph [15].

2.3 Representation using Structure and Content

Most existing approaches do not use these two dimensions (structure and content) because of its complexity. However, for best results in the later stages of the TR (e.g. clustering, classification), it is essential to use both [16]. Here are some works in the literature. A first and easy option is to mix in a VSM representation [17] the content and document tags. In [16] are used Close Frequent Sub-trees in charge of processing the document structure and then perform a preprocessing to the contents of documents. Other work carried out extensions to the VSM representation, called C-VSM and SLVM [17, 18]. In both forms for each document a matrix M_{ext} where e is the number of labels and t the number of terms is implemented, each cell will contain the frequency of each term t_i in the label e_j. C-VSM presented the "low contribution" problem to ignore the semantic relationship between different elements and SLVM not taking into account the relationship between common elements can present the "over contribution" problem. In order to eliminate these difficulties [13] proposed the Proportional Transportation Similarity, working with weighted comparisons according to the similarity of the items to compare in two documents.

3 New Textual Representation Weighting Content as related to Structure Position

Information in XML documents are in semi-structured format, so that the textual representation is essential for further processing. In this work we have selected the VSM representation [3], which will change the way of calculating the frequency of terms in each document. The modification proposed in this paper follow the criterion that a term has more or less important for comparing two documents, depending on the place it occupies within them.

That is, given three documents d_1, d_2, d_3 and the words w_1, w_2, ..., w_n, where w_1, ..., w_k, $k<n$, are common to d_1 and d_2 and are present in important parts of documents (e.g. abstract, keywords), and w_{k+1}, ..., w_n are common to d_1 and d_3, but are present in less

important sections of these, the relationship between the documents d_1 and d_2 is stronger than there between d_1 and d_3, because since their common words belong to key parts of the document, the information from these two documents is common significantly compared to the documents d_1 and d_3.

The Textual Representation referred to in this paper has four main modules: documents corpus transformation, term extraction, dimensionality reduction, matrix normalization and weighting, the next subsections will describe these.

3.1 Corpus Transformation

To do the TR, the input is a set of tokens of words obtained in a process of Information Retrieval, these tokens will be used to generate significant features (index terms). The first step in the corpus transformation can process XML documents, identifying in which SU is present a given content. Second, the resulting sequence of tokens is transformed converting all letters to capital letter, removing punctuation marks at the end of tokens, ignoring tokens containing alphanumeric characters, and substituting contractions by their full expressions [19].

3.2 Term Extraction

This submodule starts from a tokens sequence and produce an index terms sequence based on these tokens. This paper performs a lexical analysis of texts, identifying simple words like features. Thus, the statistical plane of the texts is basically exploited and the sequence of appearance of words in a document is not considered (bag-of-words model) [20], but take into account in which SU is present the word.

In the original VSM representation each document d_j is a vector of term frequencies $d_{tf} = (tf_d(t_1), ..., tf_d(t_m))^T$, where $tf_d(t)$ denotes the term t appearing frequency in the document d. In this proposal, as mentioned above, is takes into account documents' structure, so that the frequency ($tf_d(t)$) will be weighted, depending on what SU the analyzed token occupies, being defined for a token t_i in a document d_j as shown in Equation 1, where n is the number of SU present in d, tf_{tk} is the frequency of t in the SU k, and w_{kd} is the weight the SU k in document d

$$tf_d(t) = \sum_{k=1}^{n} (w_{kd} * tf_{tk}) \quad (1)$$

In Equation 2 in shown how to perform the calculation of the each SU k in each document d; here Lk is the length of k, Ld is the length of the document d and p is a parameter that gives a degree of freedom to estimate weight.

$$w_{kd} = \left(e^{(-L_k/L_d)}\right)^p \tag{2}$$

3.3 Dimensionality Reduction

This submodule reduced the representation dimensionality, eliminating stop-words, selecting all features whose score is above or below of a threshold, or the *m* best features, considering mainly I and II quality terms expressions [21] to calculate the term quality. In addition, the spelling is homogenized and words are reduced to root form [22].

3.4 Normalization and Weighting Matrix

At this TR stage a weighted vector is generated for any document, term frequencies vector based. In the proposed implementation scheme is used TF-IDF [21] to weigh the matrix values and is normalized by dividing the terms frequency by the documents length, see equations 3 and 4.

$$tfidf(t) = tf(t) * idf(t) \tag{3}$$

$$idf(t) = \log\frac{n}{n(t)} \tag{4}$$

Finally, Figure 1 shows the schematic representation of the text corpus.

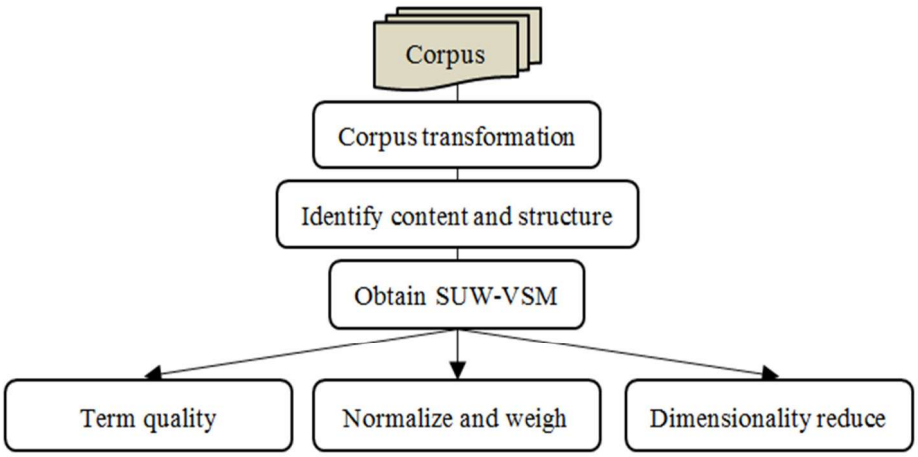

Fig. 1. Textual corpus representation scheme

4 Application of the Technique Implemented in a Management System for Scientific Papers

In [12], the System for Retrieved Research Papers Management using Lucene (GARLucene) is introduced following a general scheme that has four general modules: information retrieval or textual corpora specification process, textual corpus obtained representation, documents clustering and textual cluster obtained validation. This system uses the advantages of Lius[2] and Lucene[3] for indexing and retrieving textual information.

GARLucene in its original version manipulates XML documents but does not exploit their structure. In this paper, it is reported how it was added to GARLucene a textual representation form described in Section 3. This addition increases the GARLucene potential, since the better documents are represented, better cluster results.

4.1 Main Action to Incorporate New Textual Representation to GARLucene

For implementing this variant of XML document TR in GARLucene the following major actions were performed. Documents were indexed with Lucene library that allows incremental index creation, search and information retrieval indexing. To create the index is firstly used JDOM Java API, designed to work with XML documents to identify each SU in the documents, that was later provided as the Lucene fields and facilitate the creation of indexes. GARLucene reused largely Lucene facilities for TR. The first phase of the transformation is done in the indexing and retrieval.

Lucene allows the retrieved collection VSM representation, primarily through the *StandardAnalyzer* class that implements *StandardFilter* to normalize extracted tokens, *LowerCaseFilter* to lowercase tokens and *StopFilter*[4] to remove stop-words. Additionally, *Analyzer* allows obtaining words roots through heuristics, and treats the synonymy and polysemy. TR in GARLucene was enriched by adding the filtering methods for the feature selection calculating Quality Terms I and II [21] expressions. GARLucene implemented three variants of divisive hierarchical clustering algorithm using the edges betweenness

4.2 Clustering Algorithm based on Differential Betweenness

The clustering algorithm based on Differential Betweenness [12], parts from the proposal defined by [23] to use with the differential intermediation, achieving better results than those Newman obtained, since the good properties of the measurement are inherited.

---

[2] http://sourceforge.net/projects/lius

[3] http://lucene.apache.org

[4] Use a small stop-words list, enriched in this research.

Differential Betweenness and Cosine Similarity
When applying DB to textual domains a graph representation can be used where the interaction between two documents (edges' weight) is expressed in terms of how similar they are. Areas of high similarity values involve highly interconnected nodes. Generally, documents in the same cluster are more similar than documents in different cluster.

In [12] is illustrated the utility of the DB in the bridges detection between clusters, to cluster documents where Cosine similarity expresses the interrelationships between them. This way of calculating the edges centralities discover the bridges between clusters because, unlike the cosine similarity, it is able to exploit the graph topological properties.

Clustering Algorithm Based on the Similarity Matrix
This section show the clustering algorithm based on the concept of DB proposed by [12], see Figure 2. In [12] is described step by step this algorithm.

1. Obtaiment of the similarity graph.
2. Calculation of the weighted differential betweenness matrix.
3. Estimation of the edges to be eliminated.
4. Determination of the kernels of clustering by means of the extraction of the connected components.
5. Classification of the nodes not belonging to the kernels.

Fig. 2. Clustering algorithm based in Differential Betweenness.

5 Experimental Results

This section aims to illustrate how much are improved the results of clustering based on DB when using the textual representation proposed, with respect to the results obtained when using the classical VSM representation.

5.1 Definition of Case Studies and Tools used

For achieve the experiments two case studies were defined, the first consists of 15 corpora formed as subsets of documents in the Biomed[5] collection, the second case study has 10 corpora of XML documents from papers recovered from the ICT[6] site of the Centro de Estudios de Informática (Centre of Studies of Informatics), in the Universidad Central

[5] Bioinformatics and Medical papers http://www.biomedcentral.com/info/about/datamining/
[6] http://ict.cei.uclv.edu.cu

"Marta Abreu" de Las Villas (Central University of Las Villas). Table 1 describes these corpora (case study 1, corpus 1 to corpus 15; case study 2, corpus 16 to corpus 25).

5.2 Experiments Design and Implementation

Since we had the reference classification of textual collections considered in the experiment, we selected Overall F-measure (OFM) measurement for the comparative study of the clustering results with both TR. Equation 5 shows the general expression of OFM; equation 6 shows the F-measure calculation, that combines both expressions Precision (Pr) and Recall (Re) (see Equation 7), considering the real threshold $\alpha \in [0.1]$. Here n_{ij} is the number of objects that are in both class i and cluster j, n_j is number of objects in the cluster j, n_i is the number of objects in the class i, n is the count of clustered objects and k is the count of reference classes.

Table 1. The description of case studies.

Corpus	Document count	Class count	Reference classification
1	54	2	Cystic Fibrosis, Diabetes Mellitus
2	31	2	Cystic Fibrosis, Lung Cancer
3	26	2	Cystic Fibrosis, Microarray
4	28	2	Cystic Fibrosis, Genetic Therapy
5	42	2	Cystic Fibrosis, HIV
6	53	2	Diabetes Mellitus, Lung Cancer
7	48	2	Diabetes Mellitus, Microarray
8	50	2	Diabetes Mellitus, Genetic Therapy
9	64	2	Diabetes Mellitus, HIV
10	25	2	Lung Cancer, Microarray
11	27	2	Lung Cancer, Genetic Therapy
12	41	2	Lung Cancer, HIV
13	22	2	Microarray, Genetic Therapy
14	36	2	Microarray, HIV
15	38	2	Genetic Therapy, HIV
16	37	2	Clustering, Fuzzy Logic
17	37	2	Clustering, Association Rules
18	37	2	Clustering, Rough Set
19	41	2	Clustering, SVM
20	30	2	Fuzzy Logic, Association Rules
21	30	2	Fuzzy Logic, Rough Set
22	34	2	Fuzzy Logic, SVM
23	30	2	Association Rules, Rough Set
24	34	2	Association Rules, SVM
25	34	2	Rough Set, SVM

$$Overall\ F-Measure=\sum_{i=1}^{k}\frac{n_i}{n}max\{F-Measure(i,j)\} \tag{5}$$

$$F-Measure(i,j)=\frac{1}{\alpha(1/\Pr(i,j))+(1-\alpha)(1/Re(i,j))} \tag{6}$$

$$\Pr(i,j)=n_{ij}/n_j \qquad \mathrm{Re}(i,j)=n_{ij}/n_i \tag{7}$$

Table 2 shows the results of applying OFM to the results of clustering based on the DB when the selected collections were represented with classic VSM and the variant propose in this paper.

Table 2. The comparison of the new and original TR by OFM applied to clustering.

Corpus	OFM, classic representation	OFM, new representation
1	0,710	0,710
2	0,660	0,659
3	0,670	0,670
4	0,681	0,629
5	0,676	0,676
6	0,716	0,716
7	0,762	0,851
8	0,757	0,741
9	0,675	0,675
10	0,660	0,668
11	0,678	0,660
12	0,663	0,680
13	0,675	0,754
14	0,715	0,715
15	0,606	0,688
16	0,684	0,870
17	0,674	0,946
18	0,674	0,973
19	0,959	0,976
20	0,674	0,967
21	0,665	0,659
22	0,736	1,000
23	0,864	1,000
24	0,703	0,971
25	0,858	1,000

After obtaining the results of the OFM for the algorithm applied to each collection, statistical tests were performed to compare and analyze the significance level and behavior of the two variants of TR analyzed. In this sense, were performed to compare the algorithms non parametric tests for two related samples, using the Wilcoxon test, see Table 3 and Table 4.

For interpreting the results was considered:

- Highly significant, a significance less than 0.01,
- Significant, a result of significance less than 0.05 and greater than 0.01,
- Moderately significant, a result less than 0.1 and greater than 0.05,
- Not significant, a result greater than 0.1.

Table 3. Ranks of results.

		N	**Mean Rank**	**Sum of Ranks**
Classic-New	Negative Ranks	14[a]	12,00	168,00
	Positive Ranks	5[b]	4,40	22,00
	Ties	6[c]		
	Total	25		

[a]Classic < New [b]Classic > New [c]Classic = New

Table 4. Wilcoxon test statistics of results.

	Classic-New
Z	2.938[a]
Aymp. Sig (2-tailed)	**0,003**

[a]Base on positive ranks.

In analyzing the results of the statistical test can be seen that there are highly significant differences between the two variants of textual representation, with the textual representation proposal presented in this paper that yielded the best results of clustering, considering the OFM validation measure

6 Conclusions

This paper presented a new form of textual representation of XML documents, using their structure and content. The new form of textual representation is the content based on the structure of the document, following the criterion that depending on the location (structural unit) in the presence of a term (word) within a document, you will have greater or lesser importance to decide how relevant this is in the document. The incorporation of the new form of textual representation in GARLucene has increased significantly the potential of the software for handling XML documents and extracting knowledge from

them. This new form of textual representation yields better clustering results considering the algorithm based on the Differential Betweenness, than using classical VSM representation.

References

1. Carrillo, R.M., A.L.L.: Una Representación Vectorial para Contenido de Textos en Tratamiento de Información, In: CCC-08-004. Coordinación de Ciencias Computacionales INAOE (2008)
2. Lewis, D.D.: Representation and Learning in Information Retrieval, in Department of Computer and Information Science. University of Massachusetts (1992)
3. Salton, G., Wong, A., and Yang C.S.: A Vector Space Model for Information Retrieval. Journal of the ASIS, 18(11): pp. 613-620 (1975)
4. García-Plaza A.P., Víctor Fresno, R.M.: Una Representación Basada en Lógica Borrosa para el Clustering de páginas web con Mapas Auto-Organizativos, In: Procesamiento del Lenguaje Natural. 79-86 (2009)
5. Bakus, J.H., M.F.; Kamel, M.: A SOM-based document clustering using phrases, In: 9th International Conference on Neural Information Processing ICONIP 2002. pp. 2212-2216 (2002)
6. Liu, Y., Wang, X., Wu, C.: ConSOM: A conceptional self-organizing map model for text clustering. Neurocomputing. 71(4-6): pp. 857-862. (2008)
7. Theodore Dalamagas, T.C., Klaas-Jan Winkel, Timos Sellis: A Methodology for Clustering XML Documents by Structure. Information Systems (2006)
8. Abiteboul, S., Querying semi-structured data. Proceedings of the ICDT Conference, Delphi, Greece (1997)
9. Guerrini, G.M.M., Sanz, I.: An Overview of Similarity Measures for Clustering XML Documents. (2006)
10. Tien T., R.N.: Evaluating the Performance of XML Document Clustering by Structure only.
11. Kruse, R., Döring C., Lesor M.J.: Fundamentals of Fuzzy Clustering, in Advances in Fuzzy Clustering and its Applications. Oliveira, J.V.D., Pedrycz, W. Editors. John Wiley and Sons: East Sussex, England. pp. 3-27 (2007)
12. Arco, L.: Agrupamiento basado en la intermediación diferencial y su valoración utilizando la teoría de los conjuntos aproximados, In: Ciencias de la Computación. Universidad Central "Marta Abreu" de Las Villas: Santa Clara, Villa Clara. pp. 187. (2009)
13. Wan, X., Yang, J.: Using Proportional Transportation Similarity with Learned Element Semantics for XML Document Clustering. International World Wide Web Conference Committee (2006).
14. Flesca, S., et al.: Fast detection of XML structural similarities. IEEE Trans. Knowl. Data Engin. 7(2): pp. 160-175 (2005)
15. Chawathe, S.S.: Comparing Hierarchical Data in External Memory. In: In Proceedings of International Conference on Very Large Databases (1999)
16. Kutty, S., et al.: Combining the structure and content of XML documents for clustering using frequent subtrees. INEX, pp. 391-401 (2008)
17. Doucet, A., AhonenMyka, H.: Naive clustering of a large XML document collection. INEX. pp. 84-89 (2002)
18. Yang, W., Chen, X.O.: A semi-structured document model for text mining. Journal of Computer Science and Technology, 17(5): pp. 603-610 (2002)

19. Lanquillon, C.: Enhancing Text Classification to Improve Information Filtering, in Research Group Neural Networks and Fuzzy Systems. 2001, University of Magdeburg "Otto von Guericke": Magdeburg. pp. 231. (2001)
20. Lewis, D.D., Ringuette M.:. A comparison of two learning algorithms for text classification. In: Third Annual Symposium on Document Analysis and Information Retrieval. University of Nevada, Las Vegas. (1994)
21. Berry, M.W.: Survey of Text mining: Clustering, Classification, and Retrieval. New York, NY, USA: Springer Verlag. (2004)
22. Frakes, W.B., Baeza-Yates, R.: Information Retrieval. Data Structure & Algorithms. New York: Prentice Hall. (1992)
23. Newman, M.E.J.: Analysis of weighted networks. Physical Review E. 70(52): pp. 056131. (2004)