Specification Design for an XML Mining Configurable Application

Cristal Karina Galindo Durán, Mihaela Juganaru-Mathieu and Héctor Javier Vázquez

***Abstract*—This work presents a methodology for XML mining centered on the process of extracting document features related to structure and content. This information is used to obtain similarity measures and to cluster XML documents. A conceptual framework is proposed to design an application with the primary goal of implementing a modular and easily configurable tool for mining large XML document collections.**

***Index Terms*—conceptual framework, clustering, modularity, XML mining, UML.**

I. Introduction

CUrrently XML [1], [2] plays a very important role in the development of applications in which textual information is abundant, because it allows for storage of structured and/or semi-structured information [3] independently of any final presentation. For situations in which textual information is more abundant than numbers, traditional techniques of information storage and retrieval, such as relational databases, are poorly adapted. XML also offers compatibility between different systems and platforms and in different applications; it provides the ability to share information in a reliable and easy manner; it is the basis for the development of online applications and e-commerce; and it can be used for creating collections of documents.

However, direct use of an XML collection is difficult because it requires an understanding of technologies such as XSLT (eXtensible Stylesheet Language Transformations) and CSS (Cascading Style Sheets). To manage information stored in XML and to extract information not explicit in the collections of data, research has been undertaken on a process known as "XML mining"

The aim of the present work is to propose a framework for the design of a modular and configurable application for XML mining, in particular for document clustering in large collections such as INEX[1] (INitiative for the Evaluation of XML Retreival), composed of semi-structured textual data. This framework is intended to lead to development of a modular and configurable application–modular in the sense that it will be easy to change modules, functions or algorithms and configurable in that it will allow the user to:

- mine various large collections;
- choose different attributes about structure, content or both;
- count frequencies or count only the presence/absence of structural or content features; and
- choose different mining algorithms.

In this last case it will enable, depending of the counting method, the use of different metrics of similarity, generate different similarity matrices, and permit change of the clustering algorithms. The first part of this paper presents a brief description of XML and an overview of XML mining. Then a conceptual design is proposed. The second part of this paper uses the conceptual diagram as a basis for implementation of the application, highlighting the use of case diagrams, classes structure, and modules of the application.

II. XML Description, Collections and Mining

XML is a standard meta-language that uses extensible tags and was developed by the World Wide Web Consortium. It is a simplification and adaptation of SGML (Standard Generalized Markup Language) and defines the specific language grammar (just as HTML is itself a language defined by SGML). Therefore, XML is not a programming language but is a language of description of the information and also a way to define other languages (XHTML, SVG, MathML) for different needs. Some advantages of using XML are:

- It is easily processable (data management) by both people and by applications.
- Content and presentation are independent.
- It is designed to be used in any language or with any alphabet.
- Parsing is easy because of the strict rules governing composition of a document.
- It uses a hierarchical structure.
- The number of names of tags is unlimited.
- There are powerful tools for linking (XLink) to other XML or other type of documents.

Considering these advantages, XML is an ideal storage format for collections of text and data containing links to other documents (within the same collection or in another collection) and references to images and other multimedia files. For example, INEX provides a large collection of articles from Wikipedia in 2007, which focuses primarily on XML documents. It currently consists of two collections: a

Manuscript received December 20, 2010; revised February 07, 2011. The authors acknowledge the financial support given to the Promep group Analysis and Information Management and to the Universidad Autónoma Metropolitana, Unidad Azcapotzalco (UAM-A) for the financial support given to the project, Study and Systems Modelling, Number: 2270217 División de Ciencias Básicas e Ingeniería, UAM A. C. K. Galindo Durán acknowledges the Consejo Mexiquense de Ciencia y Tecnología (COMECyT) for the financial support received from February to July 2010.

Cristal Karina Galindo Durán, Master's Computer Science Program, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Avenida San Pablo 180, Col. Reynosa Tamaulipas, México D.F. C.P. 02200, México, email:cdgalindod@gmail.com.

Mihaela Juganaru-Mathieu, Laboratoire en Sciences et Technologies de l'Information, Institut H. Fayol, Ecole Nationale Supérieure des Mines de Saint Étienne, 158, cours Fauriel, 42023, SAINT ÉTIENNE Cedex 2, France, email: mathieu@emse.fr.

Héctor Javier Vázquez, Departamento de Sistemas, Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Avenida San Pablo 180, Col. Reynosa Tamaulipas, México D.F. C.P. 02200, México, email: hjv@correo.azc.uam.mx.

[1]http://www.inex.otago.ac.nz/

large one that has about 2.7 million documents and a smaller one that contains approximately 60,000 documents. This collection includes also bags of words that are commonly found in documents and frequencies of various structures in the form of XML tags, trees, links and names of entities.

The focus of XML mining is to develop and use techniques to extract information from XML documents. XML mining has been influenced by other fields, such as data mining, information retrieval, pattern recognition, and artificial intelligence. XML mining [4] can be defined as the process that enables discovery of new knowledge not explicit in the documents analyzed, but that is revealed once relationships are established among the contents of several structured documents.

The extraction of information is achieved by taking into account information about the structure of documents themselves, and/or their content, either textual or numerical. This does not mean that traditional techniques used for data mining in general [5] will not be useful, but it does mean that XML's multidimensional structure allows the management of information in different ways [6]. XML research has led to the development of various techniques and tools [7], [8] dealing with data frequency patterns as well as frequency patterns within parse trees.

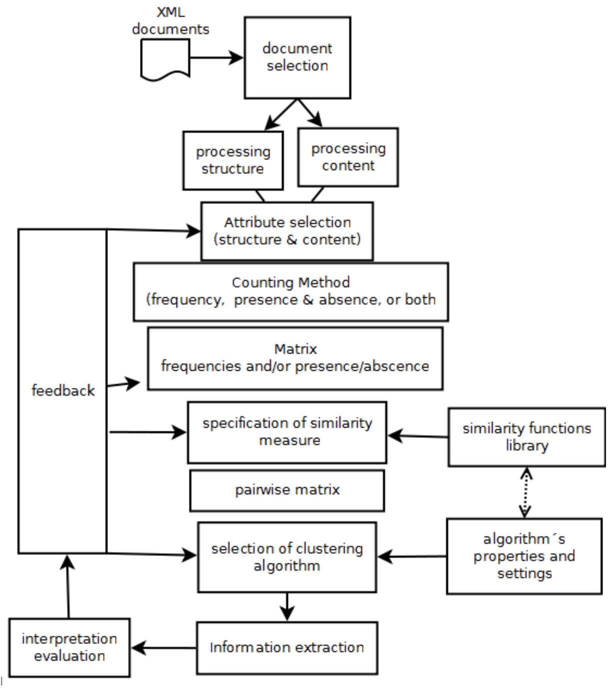
XML documents
document selection
processing structure
processing content
Attribute selection (structure & content)
Counting Method (frequency, presence & absence, or both
Matrix frequencies and/or presence/abscence
feedback
specification of similarity measure
similarity functions library
pairwise matrix
selection of clustering algorithm
algorithm´s properties and settings
interpretation evaluation
Information extraction

Fig. 1. Conceptual Model

III. Conceptual Design for XML Mining

The following is an overview of the process of extracting information from XML:

1) Selection set of XML documents.
2) Pre-processing of XML documents (discovery of the structure, identification of similar items, and/or removal of common characteristics).
3) Selecting and applying the XML mining technique, such as clustering.
4) Information extraction.
5) Interpreting, evaluating and, eventually, reconfiguring procedures.

Pre-processing of XML documents is necessary to identify their structure and content. For example, structure is revealed throught document parsing. This operation generates a parse tree of the XML document. Once the structure is known, content is identified by locating similar items and/or removing items with similar characteristics. During this process, all or a subset of attributes can be selected and counted (frequency and presence and/or absence). These counts can be integrated in a single table or organized in different count or frequency matrices [7]; for example, one frequency count matrix to register all documents and information about their structure (tags, markers, token path, depth or any another feature related to structure) and another matrix to register counts of tokens about documents content (composed mainly of text). Depending on the nature of the counts, there are different kinds of similarity measures, whether they are frequencies or binary data. For example, in the case of frequency counts, it is common to use the Euclidean, Mahalanobis, Minkowski and χ^2 distances [9]. In the case of the presence or absence of particular features, it is possible to assign more or less importance to presences than to absences: the Jaccard and Dice-Sorensen indexes are most often used for this purpose. However, if both presences and absences are considered to have the same importance then, the Sokal and Michener, Sokal and Sneath, Hamming, or Roux [9] indexes may be used. Once the metric is chosen, similarity matrices are obtained and then different cluster procedures can be applied such as expectation maximum (EM) based or Hierarchical algorithms.

The proposed application will be oriented to clustering and not to classification [10]. Clustering is the process of partitioning data into a set of groups, clusters, or subclasses. This process is performed without any information about possible groups and without examples or learning sets. Once the process finished, all data in the cluster share common traits [11], [12]. Given that the user is not directly involved until the end of the process, clustering is considered to be unsupervised classification. Conversely, in supervised classification, which is used to predict group membership for data instances, the user intervenes and monitors the process.

Once the clusters have been obtained, they can be parsed to find data structures and explore their characteristics,which can be interpreted and evaluated, and the mining process eventually can be repeated. The Figure 1 depicts this process with the goal of using it as a specification to develop a mining application.

IV. Software Design

From the description of the conceptual model, it might seem that XML mining is a linear procedure, however; the process must be guided by the user during selection of attributes, similarity measure alternatives, cluster algorithms and other settings. Therefore, a valuable, user-oriented application would be one that gives the user the ability to choose among different types of tokens, different types of similarity distances, and even different types of clustering algorithms. All these options can be applied first on a large collection or on a test collection. At the end of the process and during user interpretation of results, the application would "feed" user's knowledge into the XML mining process. The Unified Modeling Language (UML[2]) is used to present different elements of the proposed design. In the following sections, use case, class diagrams and implementation of an entity relational model to store information are presented.

A. *Use Case Diagrams*

As shown in the conceptual model, a use case for each step (eight) and a class diagram are proposed. The use case diagrams illustrate specific aspects of interactions of the user with the application. As an example, the four user case diagrams related to the most critical functionalities are:

- Use case to import XML documents: This case presents the activities to obtain the XML collection. The user connects to the INEX site, saves the list of zip files, then extracts and stores XML documents.
- Use case for pre-processing documents: Documents should be cleaned (like removing undesired features) under expert or user predefined rules. After the structure is revealed (for example, with a parse tree) and content is studied, the next step is to obtain tokens and count them.
- Use case for pairwise matrix generation: Frequency or presence/absence counts are established for tokens about content and structure. From this information, a mixed frequency matrix is obtained, pairwise distances are calculated, and the pairwise matrix is built.
- Use case for clustering: This includes all the actions to obtain clusters. Different cluster algorithms can be used, without the necessity of changing the other modules.

The remaining use case diagrams present interactions with results evaluation and their management.

B. *Class Diagram*

In the class diagram, classes are first defined for the XML collection and its documents. These two classes are directly associated with a multiplicity [1,*], given that one collection has several documents. An important class is the "Distribution_frequency" which is associated with several documents and several tokens. Tokens are the objects identified in the document and may have several categories (for example, structural or grammatical). The remaining classes are related to the mining process, which applies cluster algorithms, based on a given metric to the frequency matrix (revealed by classes: Token and Distribution_frequency).

The classes proposed are: **Collection_C, Document_C, Distribution_Frequency_C, Token_C, Criteria_C, Choice_C, Metrics_C, Apply_C, Cluster_C, and Algorithm_C.**

Figure 2 shows a static view of the class diagram, with the ten classes and their associations, proposed for this application.

The design process was separated into two packages, as follows:

- package one to obtain documents, to tokenize them and store them, and to obtain the frequency matrix

[2]http://www.omg.org/uml

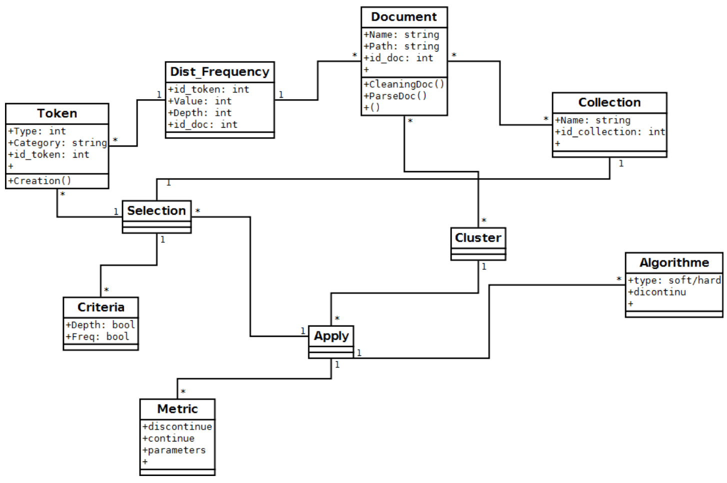

Fig. 2. Class Diagram

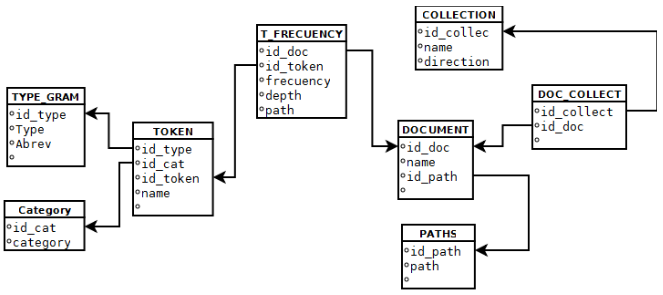

Fig. 3. Relational Model

- package two to deal with the mining process itself.

For the first package it is proposed to use a database management system (DBMS).

C. *Relational Model*

The eight tables of the relational model are presented in Figure 3, as follows:

1) The **COLLECTION** table with a tuple for each XML document collection.
2) The **DOCUMENT** table references **PATH** table and contains a tuple for each XML document.
3) The **DOC_COLLECTION** table indicates the composition of collections. It references the **COLLECTION** table and the **DOCUMENT** table.
4) The **PATH** table indicates the computer path where the XML files are physically stored.
5) The **TOKEN** table stores a unique tuple for each possible token and references the tables **TYPE_GRAM** and **CATEGORY**.
6) The **CATEGORY** table contains a tuple for each category of a token: word, tag, link or other.
7) The **TYPE_GRAM** table indicates the possible grammatical type for a token word.
8) The **T_FREQUENCY** table indicates frequency of tokens in documents and has two references: one to the **DOCUMENT** table and a second one to the **TOKEN** table.

D. Implementation

Our major decision about the first part of this application was to store, on a DBMS, all the information about documents, tokens and frequencies in a data base. The purpose is to avoid calculating all feature frequencies (structure and content) each time the user analyzes different sets of attributes. Frequency matrices and presence/absence matrices will be directly extracted with aggregated SQL queries from the tables of the database. PostgreSQL is used for the database implementation of the DBMS.

Another decision was to use a grammatical invariable generator such as Tree Tagger[3], a part of speech tagger that obtains the grammatical category and invariable form of every word [13].

The least, but not the less important, decision was to use an object oriented language (in our case Java) to implement the whole application. This allowed us to introduce and consider new distance measures or mining algorithms. Java also offers the possibility of working with various APIs for DBMS or XML, to interface with Tree Tagger or to call mining procedures or libraries written in GNU applications such as R[4] or other languages.

The first steps are to obtain the XML collection, assign a number and identify the computer path where each document is stored. This information is stored in the DOCUMENT and PATH tables.

To track and access information stored in XML documents, there are several mechanisms. Basic mechanisms are implemented in parsers such as SAX (Simple API for XML[5]) and DOM (Document Object Model[6]). SAX tracks the document sequentially in depth while DOM randomly accesses the document and builds, in memory, a tree of the XML document, permitting the document to be parsed, in depth or in breadth, or its elements to be modified. Because DOM is less complex, it is used to obtain tags, hyperlinks and paths tags. After parsing, Tree Tagger is used to obtain content tokens. From this, a token frequency matrix is created, which forms the basis of the content and structure frequency matrices. Token frequency matrix generation requires extra storage, but, as explained previously, it was decided to implement it, given that it is simple to obtain whole tokens counts, and because it later gives the user a choice of generating the frequency or the presence/absence matrix. From another perspective, this option actually saves space, because it is not necessary to save features in text form nor separate frequency matrices (structure and content) each time the user analyzes different sets of attributes.

V. Conclusion

In this paper, we presented a UML-based software model design for implementing a modular and configurable application for XML mining and clustering in a large collection of documents. The conceptual model is a framework for understanding and following user actions to obtain the information and process it. Concerning the generation of a token frequency matrix, this choice saves space and it allows, if required, to obtain one or two separate frequency matrices, one for structure and other for content, given that each token category can be easily identified in a single table.

Althought this work is not new, it highlights the importance of developing a configurable and modular application; the research works have excluded the possibility of modifying parts of the pre-processing and mining application or changing its configuration.

The first stage (package one) of development of the software application has been completed and uses Java and PostgreSQL for implementing the database management system. This implementation has been tested with 60,000 documents, and we have been able to obtain the token frequencies. These frequencies will be the basis for generation of frequency matrices, which in turn are the basis for the second package, which focuses on the mining process.

Acknowledgments

The authors acknowledge Dr. Nicolas Dominguez Vergara (former head of the Systems Department) and Dr. Silvia González Brambila (Coordinator of the Master's Computer Science Program at the UAM Azcapotzalco) for their support and encouragement. They also acknowledge Peggy Currid for her advice in copyediting.

References

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "Extensible markup language (XML) 1.0 (fifth edition), w3c recommendation," Available: http://www.w3.org/TR/REC-xml/, 2008.

[2] E. R. Harold and W. S. Means, *XML in a Nutshell, Third Edition*. O'Reilly Media, 2004.

[3] S. Abiteboul, P. Buneman, and D. Suciu, *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2001.

[4] L. Candillier, L. Denoyer, P. Gallinari, M. Rousset, A. Termier, and A. Vercoustre, "Mining XML documents," in *Data Mining Patterns: New Methods and Applications*, P. Poncelet, F. Masseglia, and M. Teisseire, Eds. Information Science Reference, 2007, ch. 8, pp. 198–219.

[5] J. Hernández-Orallo, M.-J. Ramírez-Quintana, and C. Ferri-Ramírez, *Introducción a la Minería de Datos*. Pearson - Prentice Hall, 2004.

[6] T. Tran, R. Nayak, and P. Bruza, "Document clustering using incremental and pairwise approaches," in *Focused Access to XML Documents 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007*. Springer Verlag, 2008, pp. 222–233.

[7] ——, "Combining structure and content similarities for XML document clustering," in *7th Australasian Data Mining Conference*, 2008.

[8] J.-W. Lee and S.-S. Park, "Computing similarity between XML documents for XML mining," in *Engineering Knowledge in the Age of the Semantic Web, 14th International Conference, EKAW 2004*. Springer Verlag, 2004, pp. 492–493.

[9] R. Maurice, *Algotithmes de Classification*. Ed. Masson, Paris, France, 1985.

[10] "Oracle text application developer's guide release 1 (10.1)," Available: http://www.stanford.edu/dept/itss/docs/oracle/10g/text.101/b10729/classify.htm, 2010.

[11] M. Garre, J.-J. Cuadrado, M. A. Sicilia, D. Rodríguez, and R. Rejas, "Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software," *Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 3, no. 1, pp. 7–22, 2007.

[12] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[13] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *New Methods In Language Processing*, D. B. Jones and H. Somers, Eds. Routledge, 1997.

[3]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
[4]http://www.r-project.org/
[5]http://www.saxproject.org/
[6]http://www.w3.org/DOM/