# xml2arff: An Automatized Tool for Data Extraction in XML Files to Use in Data Science with Weka and R

**Gláucio R. Vivian**[1], **Cristiano R. Cervi**[1]

[1]Institute of Exact Sciences (ICEG)
University de Passo Fundo (UPF) – Passo Fundo – RS – Brazil

{149293,cervi}@upf.br

***Abstract.*** *Este artigo relata o desenvolvimento de uma ferramenta para auxiliar os pesquisadores de Data Science na extração de dados em arquivos XML para os softwares Weka e R. Utilizamos a linguagem de consulta XQuery para recuperarmos as informações. A ferramenta proposta executa um processo automatizado de análise dos dados. Finalmente existe a possibilidade de exportação dos dados em formatos nativos para softwares como Weka e R. Essa automatização resulta em uma redução significativa de tempo entre a recuperação dos dados e seu processamento quando comparado ao processamento manual.*

***Resumo.*** *This short paper reports the development of a tool to assist researchers at Science Data in extraction of data from XML files to software Weka and R. We use the query language XQuery to recover the information. The proposed tool performs an automated process of data analysis. Finally there is the possibility of exporting data in native format for softwares such Weka and R. This automation results in a significant reduction of time between data retrieval and processing when compared to manual processing.*

## 1. Introduction

The storage and exchange data in XML (Extensive Markup Language) is increasingly common in the daily lives of researchers. This file format is defined and recommended by the W3C (World Wide Web Consortium). There are numerous technologies designed to interact with the XML format, each with a specific purpose. Among them the language XQuery[1] allows consultations directly in XML files. His power of expression is equivalent to the SQL language for relational database.

In Data Science there are several software for data analysis. For studies with statistical characteristics we use the R language [Jaffar et al. 1992] [Gentleman et al. 2009]. In the case of data mining and machine learning, the Weka[Hall et al. 2009] tool is widely used to present several algorithms options. These tools are characterized by requiring the entry of data in a predefined format. There are some data import routines, but that require user interaction to run and often by their repetitive feature ends up making the onerous task.

In the work [Hornik et al. 2009] exposes the need for integration between the R statistical software with Weka. This integration makes it possible to use machine learning

[1] http://www.w3.org/TR/xquery/

techniques / knowledge discovery. The authors present the RWEKA[Hornik et al. 2007] package R. The same enables the use of Weka algorithms through their available functional interfaces. The RWEKA has the disadvantage to access only a subset of Weka resources and needs to be updated periodically as the Weka evolves.

The aim of this paper is to present a tool that allows automated extraction of data in XML files to the native format of the Weka software and R used in Data Science analyzes. This process simplifies the exchange of information significantly reducing the time spent for such a task.

This short paper is organized by the follow: Section 2 present of tool. Section 3 present the experiments and results. Finally in the Section 4 the final considerations and future works.

## 2. The xml2arff tool

The xml2arff tool (read: XML to ARFF) enables the conversion of data into XML files to the native format of the Data Science tools. The tool uses the XQuery language as the default for queries. Consultations on local files using the Saxon[2] library. If the files are in another repository is also possible to perform client / server connection with baseX[3] Server using the TCP / IP protocol. The query should be designed to return data not hierarchically, ie all the attributes in a row. The result of the query is processed by automated algorithm. This step seeks to identify the characteristics of attributes (type and frequency) to subsequently generate the metadata and formatting attributes to the target format (CSV or ARFF). The result is presented to the user for evaluation, which can freely change the nomenclature and type attribute if you wish. Figure 1 you can view all operating steps of the proposed tool. Figure 2 can see the class

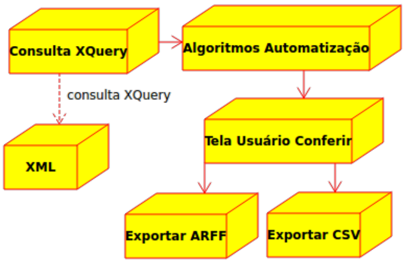


**Figure 1. Steps of process**

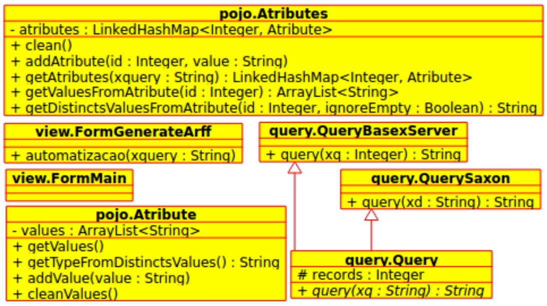

pojo.Atributes
- atributes : LinkedHashMap<Integer, Atribute>
+ clean()
+ addAtribute(id : Integer, value : String)
+ getAtributes(xquery : String) : LinkedHashMap<Integer, Atribute>
+ getValuesFromAtribute(id : Integer) : ArrayList<String>
+ getDistinctsValuesFromAtribute(id : Integer, ignoreEmpty : Boolean) : String
view.FormGenerateArff
+ automatizacao(xquery : String)
query.QueryBasexServer
+ query(xq : Integer) : String
view.FormMain
query.QuerySaxon
+ query(xd : String) : String
pojo.Atribute
- values : ArrayList<String>
+ getValues()
+ getTypeFromDistinctsValues() : String
+ addValue(value : String)
+ cleanValues()
query.Query
# records : Integer
+ query(xq : String) : String

**Figure 2. Diagrams of classes**

The format of native Weka file is the ARFF (Attribute Relation File Format). In this format attributes can be of three types. The numerical data are represented as numeric. If data has high repetition frequency, ie with few different values, the type must be rated. This differentiation allows the nominal Weka optimize the use of computing resources. Otherwise, the data is formatted as a string. The format also provides for the existence of empty value for the attribute. For more details about the ARFF format see [Holmes et al. 1994]. In the case of R software, the standard for the files is CSV (Comma Separated Value). In this format the attributes are separated by a comma. Usually the first line of the file has the naming attributes. The decimal separator must be the point. The

[2] http://saxon.sourceforge.net

[3] http://basex.org

strings must be lapped with double quotes.

## 2.1. Algorithm of Automation

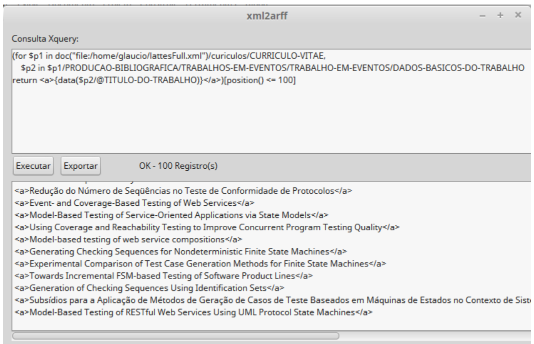
Automation step is performed by a specific algorithm. It performs data analysis for each query attribute. From this, we identify the data type and frequency. This information will be useful to generate the metadata and format the values according to data type. The asymptotic complexity is $O(attributes * values)$. The algorithm was implemented in Java following the UML documentation figure 2. Below you can see the algorithm in pseudocode.

**Algorithm 1** Algorithm of Automation

```
function AUTOMATION(xquery, maxDistictValuesToNominal)
    $atributes \leftarrow Atributes.getAtributes(xquery)$
    for all $i\ from\ atributes.size()$ do
        $number \leftarrow 0$
        $valoresUnicos.clean()$
        for all $j\ from\ atributes.getValues().get(i).size()$ do
            $value \leftarrow atributes.get(i).getValues().get(j)$
            if $isNumber(value)$ then
                $number \leftarrow number + 1$
            end if
            if $valoresUnicos.naoContem(value)$ then
                $valoresUnicos.adicionar(value)$
            end if
        end for
        if $number = atributes.get(i).getValues().size()$ then
            $atributes.get(i).setType(number)$
        else if $valoresUnicos.size() <= maxDistictValuesToNominal$ then
            $atributes.get(i).setType(nominal)$
        else
            $atributes.get(i).setType(string)$
        end if
    end for
end function
```

## 2.2. The GUI of tool

The GUI(Graphical User Interface) was built using the graphical library Swing of Java. In figure 3 you can view the screen of an XQuery query and the result of it. Figure 4 view the result of the automation process and the options for exports to ARFF and CSV formats.



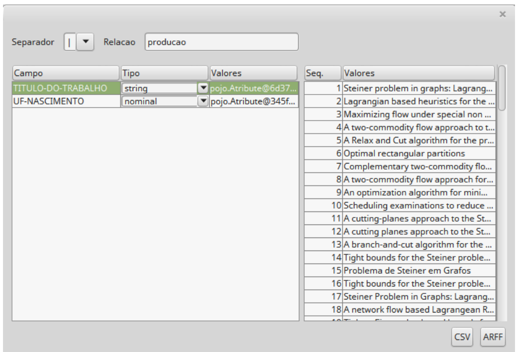
Consulta Xquery:

```
(for $p1 in doc("file:/home/glaucio/lattesFull.xml")/curiculos/CURRICULO-VITAE,
   $p2 in $p1/PRODUCAO-BIBLIOGRAFICA/TRABALHOS-EM-EVENTOS/TRABALHO-EM-EVENTOS/DADOS-BASICOS-DO-TRABALHO
return <a>{data($p2/@TITULO-DO-TRABALHO)}</a>)[position() <= 100]
```

OK - 100 Registro(s)

```
<a>Redução do Número de Seqüências no Teste de Conformidade de Protocolos</a>
<a>Event- and Coverage-Based Testing of Web Services</a>
<a>Model-Based Testing of Service-Oriented Applications via State Models</a>
<a>Using Coverage and Reachability Testing to Improve Concurrent Program Testing Quality</a>
<a>Model-based testing of web service compositions</a>
<a>Generating Checking Sequences for Nondeterministic Finite State Machines</a>
<a>Experimental Comparison of Test Case Generation Methods for Finite State Machines</a>
<a>Towards Incremental FSM-based Testing of Software Product Lines</a>
<a>Generation of Checking Sequences Using Identification Sets</a>
<a>Subsídios para a Aplicação de Métodos de Geração de Casos de Teste Baseados em Máquinas de Estados no Contexto de Sist
<a>Model-Based Testing of RESTful Web Services Using UML Protocol State Machines</a>
```

**Figure 3. Screen of query**



Separador | Relacao producao

| Campo | Tipo | Valores |
|---|---|---|
| TITULO-DO-TRABALHO | string | pojo.Atribute@6d37... |
| UF-NASCIMENTO | nominal | pojo.Atribute@345f... |

| Seq. | Valores |
|---|---|
| 1 | Steiner problem in graphs: Lagrang... |
| 2 | Lagrangian based heuristics for the ... |
| 3 | Maximizing flow under special non ... |
| 4 | A two-commodity flow approach to t... |
| 5 | A Relax and Cut algorithm for the pr... |
| 6 | Optimal rectangular partitions |
| 7 | Complementary two-commodity flo... |
| 8 | A two-commodity flow approach for... |
| 9 | An optimization algorithm for mini... |
| 10 | Scheduling examinations to reduce ... |
| 11 | A cutting-planes approach to the St... |
| 12 | A cutting planes approach to the St... |
| 13 | A branch-and-cut algorithm for the ... |
| 14 | Tight bounds for the Steiner proble... |
| 15 | Problema de Steiner em Grafos |
| 16 | Tight bounds for the Steiner proble... |
| 17 | Steiner Problem in Graphs: Lagrang... |
| 18 | A network flow based Lagrangean R... |

**Figure 4. Screen of automation**

## 3. Experiments and Results

We conducted a series of tests to measure the average time data type recognition execution / frequency and average export time for ARFF and CSV format. The tests were performed on a notebook with an Intel Core i5 third generation 4-core 2.9 Ghz, 8 GB of RAM and SSD HD 240GB. Figure 5 you can view the results for 100, 1000 and 10000 records. At each stage was tested at 1, 5, 10, 15 and 20 attributes. Figure 6 you can view part of the generated ARFF file.

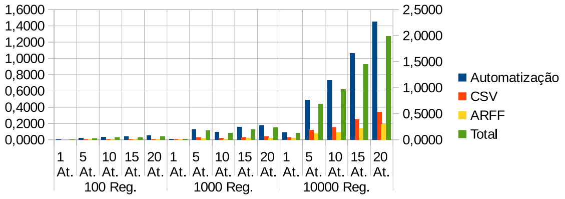
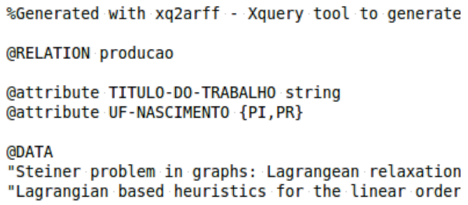


**Figure 5. Time of each step**



```
%Generated with xq2arff - Xquery tool to generate

@RELATION producao

@attribute TITULO-DO-TRABALHO string
@attribute UF-NASCIMENTO {PI,PR}

@DATA
"Steiner problem in graphs: Lagrangean relaxation
"Lagrangian based heuristics for the linear order
```

**Figure 6. The arff file generated**

It is observed that even in the worst case (20 attributes and 10000 records) the total time was around 2 seconds (right scale), thus satisfying the tool has run time compared to the extraction / manual formatting. The validation of the generated file directly was held in Weka and R software.

## 4. Final Considerations and Future Works

The xml2arff tool is at a preliminary stage. Yet it already allows the reduction of the data conversion time in XML format to the ARFF and CSV formats when compared to manual formatting. As future work if you want to expand the number of supported formats (Matlab, Scilab and Sci2). We intend to make the application available as open source software and multi language support.

## References

Gentleman, R., Ihaka, R., Bates, D., et al. (2009). The r project for statistical computing. *URL: http://www. r-project. org/254.*

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Holmes, G., Donkin, A., and Witten, I. H. (1994). Weka: A machine learning workbench. In *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*, pages 357–361. IEEE.

Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets weka. *Computational Statistics*, 24(2):225–232.

Hornik, K., Zeileis, A., Hothorn, T., and Buchta, C. (2007). Rweka: an r interface to weka. *R package version 0.3-4., URL http://CRAN. R-project. org/package= RWeka.*

Jaffar, J., Michaylov, S., Stuckey, P. J., and Yap, R. H. (1992). The clp (r) language and system. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 14(3):339–395.