Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 330 – 339

Procedia Computer Science

17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

Semi-structured documents mining: a review and comparison

Amina MADANI[a]*, Omar BOUSSAID[b], Djamel Eddine Zegour[c]

[a]*LRDSI Laboratory, Saad Dahlab University, BP 270 Soumaa road, Blida, Algeria*
[b]*ERIC Laboratory, Lumière lyon 2 University, 5 Pierre Mondès Avenue 9676 Bron Cedex, Lyon, France*
[c]*LCSI Laboratory, National High School of Computer Science, BP 68M Oued Smar, Algiers, Algeria*

Abstract

The number of semi-structured documents that is produced is steadily increasing. Thus, it will be essential for discovering new knowledge from them. In this survey paper, we review popular semi-structured documents mining approaches (structure alone and both structure and content). We provide a brief description of each technique as well as efficient algorithms for implementing the technique and comparing them using different comparison criteria.

Keywords: Semi-structured documents; documents mining; clustering; association; classification; structure mining; content mining

1. Introduction

Semi-structured documents play an important role in the exchange of data on the Web and elsewhere in various environments. With their continuous growth, many issues concerning the management of large semi-structured documents sources have also arisen. It is thus important to devise automatic procedures to extract useful information from them. Then, there is a great need to apply data mining techniques to retrieve and analyse vast amount of semi-structured documents. Most data mining techniques are not designed for these documents and should be at least adapted in order to deal with them.

In the recent years, XML† (*eXtensible Markup Language*) has reached a wide acceptance as the relevant standardization for representing semi-structured documents. XML documents present the advantage to have an explicit structure that facilitates their presentation and their exploitation in different contexts. They are

* Corresponding author. Tel.: +213-0557028356.
E-mail address: a_madani@esi.dz.
† http://www.w3.org/XML

doi:10.1016/j.procs.2013.09.110

becoming more common in various environments, permitting to represent jointly the textual information with the structure one.

This focus on XML documents can be extended to other types of semi-structured documents, such as RDF[‡] (*Resource Description Framework*) and OWL[§] (*Web Ontology Language*) documents. RDF describes and interchanges semantic data on the web. OWL is the new standard for ontology representation and exchange on the Internet.

Hence, the problems to land will be different and semi-structured documents mining is a very promising area to data mining. They require new efficient data mining techniques to extract knowledge characterized by documents structure and content. When dealing with semi-structured documents, it may be relevant to consider both structure and content information.

The next section gives a description of semi-structured documents. In section3 we present a state of the Art and a comparative study of semi-structured documents mining approaches. The paper is then concluded and further work is outlined.

2. Semi-structured documents

All knowledge, memorized, stocked on a support, fixed by writing or recorded by a mechanical, physical, chemical or electronic means constitutes a document [1].

A semi-structured document is a bridge between structured and unstructured data [2]. Unstructured data (also called flat data) is data that we know neither the context, nor the way information is fixed. It includes documents of mostly natural-language text, like word-processing files, e-mail, and text fields from databases or applications.

Structured data has a major regular structure based on descriptive markup [3]. It is commonly found in database management systems. In this data, we do not talk anymore about text, but rather about data. The significance of data depends on the structure in which it is registered. The order of structured data elements is generally not meaningful.

Semi-structured data arises when the source does not impose a rigid structure (such as the Web) and when data is combined from several heterogeneous data sources [4].

```
<article>
  <title>Equipment replacement under technological change</title>
  <journal>
    <name>Nav. Res. Logist.</name>
    <volume>41</volume>
    <number>1</number>
  </journal>
  <abstract>For infinite-horizon replacement economy problems it is common practice to truncate the problem at some finite horizon. We develop bounds on the error due to such a truncation. Bounds are illustrated through a numerical example (see page 148) from a real case in vehicle replacement. </abstract>
</article>
```

Fig. 1. An example of a semi-structured document

[‡] http://www.w3.org/RDF/
[§] http://www.w3.org/OWL/

Unlike traditional well-structured data whose schema is known in advance, semi-structured data does not have a fixed schema, it is self-describing. They are characterized by the presence of a flexible structure organizing their heterogeneous textual contents. The structure of semi-structured documents is often implicit, and not as rigid, as regular or complete as that found in traditional databases systems [5].

Semi-structured documents are characterized by the fact that they contain a mix of short ungrammatical (or weakly grammatical) text fragments, mark-up tags, and free texts [6]. HTML (*HyperText Markup Language*), SGML (*Standard Generalized Markup Language*), XML, RDF, RSS (Rich Site Summary), OWL, RDFS (*Resource Description Framework Schema*) and DC (*Dublin Core*) are examples of semi-structured documents.

Figure 1 shows an example of a semi-structured document in which we note that some parts are very structured like title and journal whereas others are unstructured like abstract.

Semi-structured data arises under a variety of forms for a wide range of applications such as genome databases, scientific databases, libraries of programs and more generally, digital libraries, on-line documentations, electronic commerce [5].

3. Semi-structured documents mining

Semi-structured documents have recently emerged as an important topic of study for a variety of reasons [7]:

- First, there are data sources such as the Web, which we would like to treat as databases but which cannot be constrained by a schema.
- Second, it may be desirable to have an extremely flexible format for data exchange between disparate databases.
- Third, even when dealing with structured data, it may be helpful to view it as semi-structured data (based on a self-describing schema) for the purposes of browsing.

Hence there has been increasing demand for automatic methods for extracting useful information, particularly, for discovering rules or patterns from large collection of semi-structured documents, namely, semi-structured documents mining. Semi-structured documents mining is the application and the adaptation of data mining techniques in order to take into account the specificities of semi-structured documents. It is a collective consequence of a variety of efforts including not only the data mining, but also, text mining, and recent web mining.

When dealing with semi-structured documents, according to the prior information available on the collection, it may be relevant to consider structure information alone or to consider both structure and content information.

Therefore, two main and complementary categories of approaches exist: semi-structured documents structure mining and semi-structured documents content mining. Structure mining extract knowledge from documents structure (tags) and content mining extract knowledge from documents content (text).

In this section, we describe and compare different semi-structured documents mining approaches that have been proposed. We begin by presenting a detailed description of the comparison criteria used:

- Doc (*Semi-structured document type*): Popular types of semi-structured documents as HTML, XML, RDF and OWL, are used and exchanged on the web. We specify for each approach the semi-structured document type that is used.
- Tech (*Data mining techniques*): Data mining techniques like clustering, classification and association rules has been widely used for semi-structured documents. For each approach, we investigate how data mining techniques can be adopted.
- Representation: A semi-structured document representation is a transformation of a document, in a format which is easier to understand. We present here the usual semi-structured documents representations.
- Contribution: We briefly describe key approaches by presenting their contribution as well as main and innovative ideas proposed.

- Algorithm: Traditional data mining algorithms are used for semi-structured documents. New mining algorithms are also proposed. For each approach, we mention the mining algorithm used.
- TS (*Tree Structure*): Semi-structured documents have generally a hierarchical structure. They can conceptually be interpreted as a tree structure that contains multiple pathways connected by named nodes. Some popular approaches model a semi-structured document as a tree, others ignore the tree structure of elements and words were not joined to their paths in documents.
- NO (*Nodes Order*): Order was considered a side issue in semi-structured data, but it becomes a central problem for XML [8]. We verify for each approach, within the representation of documents, if nodes elements appear in sequence or in an independent manner.
- ST (*Semantic Treatment*): Semantics is the study of meaning in language [9]. In semi-structured documents, semantic treatment (lexical not grammatical) has for goal to study the semantic relationships between words. Hence, the problem is how to distinguish between many different senses that a word may have (polysemy) or between different words that can have the same significance (synonymy)… The objective is to exploit the semantic similarity of terms composing the structure and the textual content of semi-structured documents (tags and text). The semantic treatment can use external semantic resources like ontologies, thesauruses and taxonomies. Ontologies to semi-structured documents become a major challenge in realizing the semantic data mining. In our comparison, we study if the existing approaches take into account this aspect or not.

3.1. Structure mining

The OED (*Oxford English Dictionary*), define structure as « the arrangement of, and relations between the parts or elements of something complex». A document has two structures [10]: a physical structure and a logical structure. The physical structure of a document corresponds to its presentation. It is characterized by the external organisation of data (layout). The logical structure is the internal representation of a document. It refers to the organization as seen by the author, to represent the meaning of a document. The organization of a document in chapters, sections, titles, and paragraphs concerns its logical architecture.

For example, entities represent the physical structure of an XML instance document, whereas element tags and their nesting therein dictate the logical structure.

In this section, we survey a number of approaches that propose to apply data mining techniques on structure of semi structured documents (Table 1). Authors chose to ignore the physical structure and use the logical structure of semi-structured documents as the basis for their approaches.

[11], [12] and [13] are several XML documents clustering methods. Since an XML document has a tree structure, they propose to model it as a labeled tree. [11] propose the TreeFinder algorithm that aims at discovering frequent sub-trees by searching frequent labeled trees from clusters of an XML collection.

[12] apply clustering algorithm using distance that estimate the similarity between XML trees in terms of the hierarchical relationships of their nodes. Tree structural summaries are used to improve the performance of the distance calculation and at the same time to maintain or even improve its quality. Given the structural summaries of rooted ordered labeled trees that represent XML documents, they form a fully connected graph with vertices and weighted edges. The weight of an edge corresponds to the structural distance between the vertices (trees) that this edge connects.

[13] generate the closed frequent sub-trees using the popular algorithm CMTreeMiner [14]. They generate a CD matrix representing closed frequent sub-tree distribution in documents. Using the matrix, they compute the similarity between XML documents and incrementally cluster them based on their similarity values.

Other approaches like [15], [16] and [17] propose original supervised classification techniques for XML documents which are based on structure only. Each document is viewed as a tree, represented by his tags only.

[15] propose XRules, a structural rule-based classifier in order to perform the classification task. Formally, a structural rule is an entity of the form $T \Rightarrow c_i$, where T is a structure (an XML document that can be represented

in tree format), and c_i is one of the k classes associated with T. In order to do so, they develop XMiner using TreeMiner algorithm [18]. XMiner use a set of trees to generate a set of frequent rules for each class.

Table 1. Comparison of semi-structured documents structure mining approaches.

Doc	Approach	Representation	Contribution	Algorithm	Tech	TS	NO	ST
XML	**Termier et al. 2002**	**Labeled trees**	**Discovering frequent subtrees**	**TreeFinder**	**Clustering**	✓	×	×
	Dalamagas et al. 2004	**Rooted ordered labeled trees → Connected graph**	**Use a tree structural summaries**	**Single link hierarchical algorithm**		✓	✓	×
	Kutty et al. 2007	**Rooted ordered labeled trees → CD matrix**	**Generate the closed frequent subtrees**	**CMTreeMiner / hierarchical agglomerative clustering**		✓	✓	×
XML	**Aggarwal et al. 2003**	**Rooted ordered labeled trees**	**Define structural rule based classifiers using frequent subtrees**	**XMiner**	**Classification**	✓	✓	×
	Candillier et al. 2005	**Trees → attributes-values sets**	**Use different relations between tree nodes (parent child relations, next-sibling relations and set of distinct paths…)**	**Boosted C5**		✓	×	×
	Garboni et al. 2006	**Ordered labeled trees → set of sequences**	**Extract frequent structural patterns from frequent sequences**	**Traditional sequential pattern extraction**		✓	✓	×
HTML XML	**Asai et al. 2002**	**Ordered labeled trees**	**Discovering all frequent tree patterns using rightmost expansion**	**FREQT**	**Association**	✓	✓	×
XML	**Boussaid et al. 2004**	**Boolean matrix, temporary trees → General tree → DTD**	**Preformat documents, create minimal DTD to manage tags hierarchy, and search frequent itemsets.**	**A-Priori**		✓	×	×

[16] transform each XML tree into a set of attributes-values using different relations between tree nodes (parent child relations, next-sibling relations, set of distinct paths…). The boosted C5 algorithm [19] is used to classify the attributes-values sets.

The method of [17] relies on structure discovery based on sequential pattern mining. For this purpose, they use a technique intended to transform any XML tree into a sequence of its node labels. A traditional sequential pattern extraction algorithm is then used to extract the frequent sequences from each predefined cluster. The last step of the method relies on a matching between each document of the collection and each cluster which is characterized by a set of frequent structural subsequences.

[20] and [21] are different methods for XML structure mining using association rules. [20] model semi-structured documents (XML/HTML) by labeled ordered trees. They present an efficient pattern mining algorithm called FREQT for discovering all frequent tree patterns from a large collection of labeled ordered trees. The key of their method is the notion of the *rightmost expansion*, a technique to grow a tree by attaching new nodes only on the rightmost branch of the tree. Furthermore, they show that it is sufficient to maintain only the occurrences of the rightmost leaf to efficiently implement incremental computation of frequency.

[21] propose to pre-format XML documents while collecting some information (names, paths, number of tags…). A Boolean matrix that indicates the tags composing a document is constituted. The method uses

adequate structures to manage the hierarchy between tags, the minimal DTD (*Document Type Definition*) in this case. A-Priori algorithm [22], [23] is used to find frequent item-sets and to extract association rules.

Discussion:

The majority of approaches work on XML documents structure. Mining HTML documents structure is less treated. Other types as RDF, OWL and DC are never exploited.

To represent the structure of a semi-structured document, a rooted, labeled, and/or ordered tree are generally used. *Ordered* means the order among the siblings is significant, while *labeled* means that each node in the tree is labeled with a symbol from a predefined alphabet. This representation allows preserving the hierarchical tree structure of elements, so that structural elements may include other (sub) elements.

In many approaches, node position in documents is ignored. The order of nodes ignored is justified by the fact that two documents with an identical content and a different nodes order can be semantically equivalent.

Most traditional data mining algorithms are not adapted for semi-structured documents. Then, several approaches propose to adapt them and to propose new algorithms to deal with this type of data.

An important point is ignored in all these works. It is the semantic carried by the structure of semi-structured documents. In documents collection, we can find different tags that describe the same thing or a tag denoting different concepts.

3.2. Content mining

Document content is presented under an indivisible and unstructured set of fragments like paragraphs, figures or pictures [10]. The structural elements can specify the data type of content that can include, for example, a text or a picture [24]. In XML document, content is the text between each start and end tag [25]. Semi-structured documents content mining is the advanced textual analysis, integrating particularities of documents as the semantic relations conveyed by the textual content. For example, mining for XML content is essentially mining for values (an instance of a relation) [26].

Some researchers combine structure and content mining to leverage the techniques strengths. They subsume them both under the term "mining content". We present here diverse approaches used for clustering, classification and association of semi-structured documents using structure and content information (Table 2).

[27], [28], [29], [30] and [31] are different methods used for clustering using structure and content information. They propose various transformations of documents. [27] work on OWL while the others work on XML documents. Since basic elements in OWL are classes, [27] parse an OWL document as a set of classes and a set of individuals. They analyze semantic of documents and proposes a method for computing semantic similarity. Similarity of two OWL documents is defined as weighted sum of classes sets similarity and individuals sets similarity.

[28] model XML tree by flattening their trees into their sets of sub-paths. They retain the frequency of paths and they consider sub-paths as words. Therefore, they apply standard clustering methods usually used for text.

The approach of [29] consists of merging a bag of words and a bag of tag names (text + tags) into a one vector. They apply k-means algorithm on the vector that contains the weight of document features specified by TF-IDF (*Term Frequency - Inverse Document Frequency*). They also propose to integrate a *textitude* measure to the document description process that basically measures the ratio between the weight of the structural information and the weight of the content information.

In [30], the structure of XML documents is represented as a collection of paths and the content is represented using LSK (*Latent Semantic Kernel)* [32] which is based on LSA (*Latent Semantic Analysis*) [33]. This approach enables to perform the clustering task on a large dataset by first reducing the dimension of the dataset using the incremental method and then graph clustering [34] based on a pairwise distance matrix to preserve the effectiveness of the clustering solution.

Table 2. Comparison of semi-structured documents content mining approaches.

Doc	Approach	Representation	Contribution	Algorithm	Tech	TS	NO	ST
OWL	Gao et al. 2005	Sets of classes + sets of individuals → similarity matrix	Compute simple classes similarity that considers the basic semantic, properties of classes	Hierarchical clustering algorithm	Clustering	×	×	✓
XML	Vercoustre et al. 2006	Set of paths	By considering sub-paths as words, they use simple clustering methods	Dynamic clouds		✓	×	×
	Doucet et al. 2006	Bag of tags + Bag of words → TF-IDF vector	Combine text and tag features into a single vector space using TF-IDF Propose a textitude measure	K-means		×	×	×
	Tran et al. 2007	Collection of paths + LSK Matrix → similarity Matrix	Apply incremental clustering to calculate a pairwise distance matrix and graph clustering	Incremental clustering /Graph clustering		✓	×	×
	Madani et al. 2011	Rooted ordered labeled trees → paths Matrix Bag of structure + Bag of content → Thesaurus	Combine three concepts: XML paths bag of words, and thesaurus	Constrained agglomerative algorithm		✓	✓	✓
XML	Denoyer et al. 2004	Trees → Bayesian networks	Use a generative model based on Bayesian networks and transform it into a discriminant classifier	Learning algorithm in Bayesian networks	Classification	✓	✓	×
	Knijf 2007	Labeled rooted ordered attribute trees	Discover frequent patterns from trees and select the emerging one to do classification	FAT-Miner		✓	✓	×
	Yang et al. 2007	SLVM Vector → similarity matrix	Extension of VSM to compute TF-IDF values for each word in each node of documents	SVM		✓	×	×
HTML	Taniguchi et al. 2001	Labeled ordered trees → Path Expressions	Find association pattern (pair of an expression path and a word-association pattern)	Path-Find	Association	✓	✓	×
XML	Braga et al. 2002	XML fragments → Relational table	Generate relational table from XML fragments	A-Priori		×	×	×
RDF	Jiang et al. 2006	Trees	Discovering the set of closed generalization closures instead of all frequent patterns	GP-Close		✓	×	×

Our method [31] consists of representing XML documents by a set of their paths preserving the hierarchical structure of XML tree. We exploit the semantic similarity between terms (tags and text) composing XML paths, by unifying them using a thesaurus former created. We create a thesaurus from two bags of words generated from XML documents (bag of structure and bag of content). The sets of paths are then mapped in a binary matrix. Constrained agglomerative clustering algorithm is used to organize documents into clusters based on their paths similarity. The originality of our approach is in the use of a thesaurus, to manage semantically the words presented in XML documents.

In [35], [36] and [37] works, different models was developed for classification of XML documents. [35] propose a new statistical model for the classification of semi-structured documents and consider its use for multimedia document classification. They propose a generative model able to handle both structure and content which is based on Bayesian networks. Then, they show how to transform this generative model into a discriminant classifier using the method of Fisher kernel [38].

The model FAT-CAT (*Frequent Attribute Trees based Classification*) proposed in [36] makes classification of XML documents using frequent attributes trees. FAT-miner algorithm is used to discover a set of frequent attributes trees from each class. Emerging trees are selected for each category. Each document is then transformed into a vector where each component indicates if a particular emerging tree appears into the document. Last, a classical classification algorithm is used on these vectors (Binary decision tree).

In order to represent XML documents as vectors, [37] use the SLVM *(Structured Link Vector Model*) [39]. SLVM was extended from the conventional VSM (*Vector Space Model*) [40] by incorporating document structures (represented as term-by-element matrices), referencing links (extracted based on IDREF attributes), as well as element similarity (represented as an element similarity matrix). These vectors are then used with a SVM (*Support Vector Machine*) [41] for classification.

[42], [43] and [44] extract association rules from three different types of semi-structured documents: HTML, XML and RDF. [42] introduce a method for mining HTML texts. They present Path-Find algorithm that find an interesting pattern called association rules or association paths. An association path is a pair of association patterns over tag sequences and word sequences (text sequences).

The paper of [43] presents the XMINE operator, a tool to extract XML association rules for XML documents. The operator is based on XPath, inspired by the syntax of XQuery [45] and to the work on MINE RULE [46]. XMINE can be used to specify indifferently (and simultaneously) mining tasks both on the content and on the structure of the data. The XMINE statement is processed to generate a representation of the XML mining problem as a relational table. Then, association rules are extracted through the A-priori algorithm.

[44] develop a frequent generalized pattern mining algorithm, called GP-Close, for mining generalized associations from RDF metadata. For accelerating the mining process, they employ the notion of g*eneralization closure*. A generalization closure of a pattern is an RDF statement set containing all statements in this pattern and all their generalized statements. GP-Close discovers the set of closed generalization closures instead of all frequent generalized patterns to minimize computation cost. A generalization closure is said to be closed if it does not have any superset of statements such that they are subsumed by the same set of RDF documents.

Discussion:

Until recently, most of the research on semi-structured documents mining was focused on XML documents. We note that RDF and OWL mining is less treated than XML mining. This is justified by the fact that XML allows the representation of semi-structured and hierarchal data containing not only the values of individual items but also the relationships between data items. Its structural flexibility makes it an attractive choice for representing semi-structured documents in application domains.

The semi-structured character of documents, the heterogeneity of their formats and especially of contents impose a pre-processing step to homogenize the representation and the description of these documents. For this, various representations are proposed. Most models are limited to flat document representations [28] or like a linear sequence of words (bag of words) [29]. The position of each node in the document is ignored, and the document is considered as a set of pre-processed terms, with no particular order.

Other more important representations are based fluently on trees or labeled graphs [42], [31], possibly involving attributes [36], and combining different classifiers [35].

In these representations, several approaches propose to adapt classical algorithms and to propose new algorithms to mine the content of semi-structured documents. Document structure and content semantics is

fundamental. However, few researches are devoted to their identification. Mining semi-structured documents content inherits some problems faced in text mining.

The majority of approaches ignore semantic of information inside documents. Synonymy and polysemy can cause difficulties (different label that describe the same concept or a label denoting different concepts).

4. Conclusion

With the advent of semi-structured documents, new opportunities and challenges have arisen. Our aim was to explore the problem of mining semi-structured documents as HTML, XML, RDF, OWL …

We have reviewed a summary of different approaches and models proposed in the literature according to different comparison criteria.

The goal of semi-structured documents mining is to extract useful information, particularly, for discovering rules, patterns or categories from large collections of documents.

Semi-structured documents are defined by their logical structure and their content. Actual approaches combine the structure mining and the content mining, to reach more effective results.

Mining of semi-structured documents significantly differs from structured data mining and text mining. Mining structure only task seems quite easy and simple models work very well on this task. The structure plays a minor role in determining the similarity between documents. The structure and content tasks are more challenging and encompass many different generic tasks in the document domain. An open problem is to find a good way to combine the structure and the content.

Most traditional data mining algorithms are not suitable for semi-structured documents. Several approaches propose to adapt them by using new algorithms for this data. Therefore, FCA (*Formal Concept Analysis*) is a mathematical theory that is recently used for data mining. In terms of perspectives, we will try to study the basis notions underlying *Formal Concept Analysis* and the different areas in which FCA is exploited. We will review popular data mining approaches based on FCA to propose a new approach for semi-structured documents structure/content mining based on *Formal Concept Analysis*.

References

[1] Estival, R., and Meyriat J. 1981. *La dialectique de l'écrit et du document*. Un effort de synthèse. Schéma et schématisation, pp.82–91.

[2] Trippe, B. 2001. *Do XML Editors Matter ?* Transform Magazine Volume 10 Issue 10, Pages 27- 27 PublisherCMP Media, Inc., USA.

[3] Tannier, X. 2006. *Extraction et recherche d'information en langage naturel dans les documents semi-structurés*. PhD thesis, France.

[4] Wang, K., and Liu H. 1997. *Schema Discovery for Semistructured Data*, In Proc. KDD'97, pp. 271–274.

[5] Abiteboul, S. 1997. *Querying semi-structured data*. In F. N. Afrati and P. G. Kolaitis, editors, Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, Proceedings, pages 1–18.

[6] Wong, T.L, and Lam W. 2004. Text Mining from Site Invariant and Dependent Features for Information Extraction Knowledge Adaptation, *Proc. SIAM Int'l Conf. Data Mining (SDM),* pp. 45-56.

[7] Buneman, P. 1997. Semistructured data. In Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'97), pp. 117-121. Tucson, Arizona.

[8] Suciu D. 2005. *Semistructured data and XML*, Computer and Information Science, Miscellaneous Papers ,October Volume: 3, Issue: October, Publisher: Kluwer Academic Publishers, Pages: 34–41. ISSN: 15427730. ISBN: 1402004893.

[9] Hurford, J. R. 1983. *Semantics : a coursebook*. Cambridge University Press.

[10] Géry, M. 2002. Indexation et interrogation de chemins de lecture en contexte pour la Recherche d'Information Structurée sur le Web. THÈSE de Doctorat, Université Joseph Fourier - Grenoble I.

[11] Termier, A., Rousset M. C., and Sebag M. 2002. *TreeFinder : a First Step towards XML Data Mining*. International Conference on Data Mining ICDM'02, Maebashi, pp. 450-457, Japon.

[12] Dalamagas T., Cheng T., Winkel K., and Sellis T. 2004. *Clustering XML Documents using Structural Summaries*, In Proc. of ClustWeb - International Workshop on Clustering Information over the Web in conjunction with EDBT 04, pp. 547–556, Greece.

[13] Kutty S., Tran T., Nayak R., and Li Y. 2007. *Clustering XML documents using closed frequent subtrees- a structure only based approach*, In: Workshop of the INitiative for the Evaluation of XML Retrieval.

[14] Chi, Y., Nijssen S., Muntz R. R., and Kok J. N. 2005. *Frequent Subtree Mining-An Overview*. In Fundamenta Informaticae. vol.66: pp.161-198.

[15] Aggarwal, C. C., and Zaki M. J. 2003. *XRules:An Effective Structural Classifier for XML Data,* SIGKDD 03, pp.316–325, New York.

[16] Candillier, L., Tellier I., and Torre F. 2005. *Transforming XML trees for efficient classification and clustering*. In: Workshop of the INitiative for the Evaluation of XML Retrieval INEX, pp. 469-480.

[17] Garboni, C., Masseglia F., and Trousse B. 2006. *Sequential pattern mining for structure based XML document classification*. In: INEX 2005 Workshop of the INitiative for the Evaluation of XML Retrieval, pp. 458–468.

[18] Zaki, M. J. 2002. *Efficiently Mining Frequent Trees in a Forest*. SIGKDD, pages 71-80, Edmonton, Canada.

[19] Quinlan, R. 2004. Data mining tools see5 and c5.0.

[20] Asai, T., Abe K., Kawasoe S., Arimura H., Sakamoto H. and Arikawa S. 2002. *Efficient Substructure Discovery from Large Semi-structured Data*. In Proc. of the Second SIAM International Conference on Data Mining (SDM2002), Arlington, VA, pages 158-174.

[21] Boussaid, O., Duffoux A., Lallich S., and Bentayeb F. 2004. *Fouille dans la structure de documents XML*. EGC 04, Revue des Nouvelles Technologies de l'Information, volume 2, pages 519-524, Clermont-Ferrand, France.

[22] Agrawal R., and Srikant R. 1995. *Mining sequential patterns*. 11th Int. Conf. Data Engineering, ICDE, pp 3–14. IEEE Press, 6–10.

[23] Mannila H., Toivonen H., and Inkeri Verkamo A. 1994. *Efficient algorithms for discovering association rules*. Knowledge Discovery in Databases, Papers from the 1994 AAAI Workshop (KDD'94), pages 181 – 192. AAAI Press.

[24] Piwowarski, B. 2003. *Techniques d'apprentissage pour le traitement d'informations structurées: application à la recherche d'information*. Thèse de Doctorat, Université Paris 6, 2003.

[25] Bray, T., Paoli, J. and Sperberg-McQueen C M. 1998. *Extensible Markup Language* (XML) 1.0 : World Wide Web Consortium.

[26] Nayak, R., Witt, R., and Tonev, A. 2002, *Data mining and XML document*s. In Proceedings of International Conference on Internet Computing, IC'2002, pp. 660-666, Las Vegas, Nevada.

[27] Gao, M., Liu C., and Chen F. 2005. *Clustering OWL Documents Based on Semantic Analysis*, WAIM'2005, pp.184-193.

[28] Vercoustre, A.-M., FEGAS M., Gul S., and Lechevallier Y. 2006. *A flexible structured-based representation for XML document mining*, In: Workshop of the INitiative for the Evaluation of XML Retrieval INEX, pp. 443-457.

[29] Doucet, A., and Lehtonen M. 2006. *Unsupervised classification of text-centric XML document collections*, In: INEX Workshop.

[30] Tran, T., Nayak, R,. and Bruza P. 2007. *Document clustering using incremental and pairwise approaches*. In: INEX Workshop.

[31] Madani A., Boussaid O., and Zegour D.E. 2011. *Clust-XPaths: Clustering of XML Paths*. In 7th International Conference, MLDM 2011, New York, USA. Lecture Notes in Computer Science, LNAI 6871, Springer, ISBN 978-3-642-23198-8, pages 294-305.

[32] Cristianini, N., Shawe-Taylor J., and Lodhi H. 2002. *Latent semantic kernels*. Journal of Intelligent Information Systems (JJIS) 18(2).

[33] Landauer, T.K., Foltz P.W., and Laham D.1998. *An introduction to latent semantic analysis*. Discourse Processes (25) 259–284 223.

[34] Karypis, G. 2007. *Cluto* - software for clustering high-dimensional datasets — karypis lab.

[35] Denoyer, L., and Gallinari P. 2004. *Bayesian Network Model for Semi-Structured Document Classification*. Revue Information Processing & Management, Special Issue on Bayesian Networks and Information Retrieval, Pages 807-827, Elsevier.

[36] Knijf, J. De. 2007. *FAT-CAT: Frequent Attributes Tree Based Classification*, In: Workshop of the INitiative for the Evaluation of XML Retrieval INEX, pages 485–496.

[37] Yang, J., and Zhang F. 2007. *XML document classification using extended VSM*. In: INEX Workshop.

[38] Jaakkola, T. S., Diekhans, M., and Haussler, D. 1999. *Using the Fisher kernel method to detect remote protein homologies*. In Intelligent systems for molecular biology conference (ISMB'99). Heidelberg, Germany: AAAI.

[39] Yang, J., and Xiaoou C. 2002. *A semi-structured document model for text mining*, Journal of Computer Science and Technology archive, Volume 17(5), pp 603-610.

[40] Salton, G., et MJ. McGill (1983). *Introduction to Modern information Retrieval*. McGraw-Hill.

[41] Vapnic, V. 1995. The Nature of Statistical Learning Theory. Springer, New York.

[42] Taniguchi, K., Sakamoto H., Arimura H., Shimozono S., and Arikawa S. 2001. *Mining semistructured data by path expressions*. Lecture Notes in Computer Science, 2226:378.

[43] Braga, D., Campi A., Ceri, S. Klemettinen M., and Lanzi PL. 2002. *A Tool for Extracting XML Association Rules from XML Documents*, Research paper in Proceedings of IEEE-ICTAI 2002, pp. 57-64, Washington DC, USA.

[44] Jiang, T., and Tan A. 2006. *Mining RDF metadata for generalized association rules: knowledge discovery in the semantic web era*. In Proceedings of the 15th International Conference on World Wide Web WWW '06. ACM Press, New York, pp. 951-952.

[45] Boag S., Don C., Fernndez M.F., Daniela Florescu D., Robie J., and Jrme S. 2007. *XQuery 1.0: An XML Query Language*. W3C.

[46] Meo, R., Psaila, G. and Ceri S. 1998. *An extension to SQL for mining association rules*. Data Mining and Knowledge Discovery, 2(2):195 . 224.