# Unsupervised classification of text-centric XML document collections

Antoine Doucet[1,2] and Miro Lehtonen[2]

[1] IRISA-INRIA
Campus de Beaulieu
F-35042 Rennes Cedex
France
firstname.lastname@irisa.fr
[2] Department of Computer Science
P. O. Box 68 (Gustaf Hällströmin katu 2b)
FI–00014 University of Helsinki
Finland
firstname.lastname@cs.helsinki.fi

**Abstract.** This paper addresses the problem of the unsupervised classification of text-centric XML documents. In the context of the INEX mining track 2006, we present methods to exploit the inherent structural information of XML documents in the document clustering process. Using the $k$-means algorithm, we have experimented with a couple of feature sets, to discover that a promising direction is to use structural information as a preliminary means to detect and put aside structural outliers. The improvement of the semantic-wise quality of clustering is significantly higher through this approach than through a combination of the structural and textual feature sets.
The paper also discusses the problem of the evaluation of XML clustering. Currently, in the INEX mining track, XML clustering techniques are evaluated against semantic categories. We believe there is a mismatch between the task (to exploit the document structure) and the evaluation, which disregards structural aspects. An illustration of this fact is that, over all the clustering track submissions, our text-based runs obtained the 1st rank (Wikipedia collection, out of 7) and 2nd rank (IEEE collection, out of 13).

## 1 Introduction

Document clustering has been applied to information retrieval for long. Most of this work followed the *cluster hypothesis*, which states that relevant documents tend to be highly similar to each other, and, subsequently, they tend to belong to the same clusters [1]. Clustering was then applied as pseudo-relevance feedback in order to retrieve documents that were not good direct matches to the query, but that were very similar to the best results [2]. Documents have to be clustered before querying, so as to form document taxonomies.

The quantity of data organized with an XML structure grows drastically. While XML document collections have essentially been data-centric, there are now more and more text-centric document collections. The necessity for tools to manage these collections has grown correspondingly. Clustering is one way to automatically organize very large collections into smaller homogeneous subsets.

In this paper, we explore a number of ways to exploit the structural information of XML documents so as to improve the quality of unsupervised document classification. We experiment with a number of techniques that were developed at a time when no performance evaluation framework was available. The techniques are built on top of the vector space model, which was enhanced with different types of textual and structural features. We propose to combine them at once or or in a 2-step approach, first using the structural features, and then the textual ones.

We present the corresponding results in the context of the INEX 2006 document mining track. We extend our contribution with the integration of a measure of the "textitude" of a structured document.

Because we require no document markup description (such as a document type definition — DTD), our techniques are particularly suited for experiments with several different collections, such as the ones used in the mining track 2006: the IEEE journals collection and the Wikipedia collection.

The evaluation of clustering consists of comparing automatic unsupervised classification instances to a given "gold-standard". Finding such an ideal classification is very difficult, as there may be many ways to split a document collection that are equally valid and arguable. However, we believe that the gold-standards used in the evaluation of the INEX mining track are heavily oriented towards the textual content of the document, and far less towards their structural content. Therefore, it came as no surprise that our best results were obtained by using textual features exclusively. The paper discusses this issue and elaborates on why the results should be analyzed carefully.

Section 2 covers related work. Our experimental setting and the methods to be evaluated are presented in Section 3. The performance of these techniques in the context of the INEX mining track 2006 are presented in Section 4, where we also discuss a number of issues and difficulties that are to be encountered when evaluating XML clustering. We draw conclusions and present the future directions of our work in Section 5.

## 2 Related work

Until recently, most of the research on structured document processing was focused on data-centric XML (see for example [3] and [4]). One early motivation for XML document clustering was to gather documents that were structurally similar, so as to generate a common DTD for them. Nierman and Jagadish notably proposed a tree-edit distance as a structural similarity measure of XML documents [5].

The birth of the INEX mining track in 2005 [6] provided an experimental framework very much needed for the case of text-centric document collections [7]. This triggered research at the crossroads of information retrieval, machine learning and XML databases.

There are currently two main approaches to text-centric XML document clustering. One of them is to build models naturally close to the XML tree structure, such as neural networks [8], including self-organizing maps [9]. The other approach relies on a transformation of the document structure into a flat vector space representation, before applying well-known clustering techniques [7, 10–12]. Previous work has proposed to use element labels as the structural features, and to combine them into word term features in a common $tfidf$ framework [7]. Candilier et al. [12] proposed more advanced structural features, such as parent-child or next-sibling relations. Vercoustre et al. [11] proposed to represent an XML tree by its different sub-paths, with features such as the path length, or the number of nodes it contains. An open problem for such techniques is to find a good way to combine the structural and textual features.

## 3 Procedure of the experiments

The document model we used was the vector space model. In other words, we represented documents by $N$-dimensional vectors, where $N$ is the number of document features in the collection.

Using this document model and the $k$-means algorithm, we performed our clustering experiments with various feature sets in one and two steps. We will now describe the clustering algorithm and then present the different ways we used it.

### 3.1 Clustering technique

We chose to use the $k$-means algorithm for our experiments. *K-means* is a commonly used partitional clustering technique, where $k$ is the number of desired clusters, either given as input, or determined in the loop. In the experiments, for simplicity and to allow easier comparison, we set $k$ to be equal to the desired number of classes. The algorithm relies on a initial partition of the collection that is repeatedly readjusted, until a stable solution is found.

In these experiments, we mainly decided to use $k$-means because of its linear time complexity and the simplicity of its algorithm.

Given $k$ desired clusters, $k$-means techniques provide a one-level partitioning of the dataset in linear time ($O(n)$ or $O(n(log\ n))$ where $n$ stands for the number of documents[13]). The *base* algorithm presented in Figure 1 takes the number of desired clusters as input.

### 3.2 Run descriptions

As our aim is to take into account both the semantics of the text and its structural markup, we naturally build two corresponding feature sets. Therefore, we

1. *Initialization*:
   - $k$ points are chosen as initial centroids
   - Assign each point to the closest centroid
2. *Iterate*:
   - Compute the centroid of each cluster
   - Assign each point to the closest centroid
3. *Stop condition*:
   - As soon as the centroids are stable

**Fig. 1.** Base $k$-means algorithm

need to use two baseline runs: one relies on a text-only representation, and the other on a structure-only representation.

- **Text features only**: These features are the result of a typical (unstructured) text representation. We removed the stop words, and then stemmed the remaining words using the Porter algorithm. The dimensions of the vector space are the remaining single word terms, in a canonical form.
- **Tag features only**: This representation uses the XML element labels as the dimensions of the vector space (stopwords are not removed and labels are not stemmed).

The rest of our runs are tentative ways to combine the information of unstructured text to that of the structural indicators. A simple way to do so is to combine the text and tag features into a single vector space. In other words, this approach consists in merging the bag of words and the bag of tag names. We name this representation "**text+tags**". This naïve approach serves as a baseline combination of textual and structural data. Note that we prevent the confusion between word features and tag features (the "art" element name should not be confused with the word "art").

We will now present two more advanced techniques. The first one was originally presented in 2002, at a time when no formal evaluation framework existed for XML mining experiments. We decided to revisit it in the framework of the INEX mining track. The second technique is new, it introduces a structural indicator in the context of the unsupervised classification process: the T/E measure [14].

**The 2-step approach.** Previous experiments suggested that a 2-step approach, "**tags → text**" (read "tags then text"), permits to obtain better results, by putting aside structural outliers before running the textual (semantic) classification [7]. The algorithm is described in Figure 2. To use tag features exclusively is very noisy when most of the XML elements have a purely stylistic role, as is the case in the IEEE collection. The technique presented here permits to benefit from the structural information of documents, with the internal similarity threshold as a safe-guard. Only the most cohesive tag-based clusters will be kept, while the rest of the clustering process is achieved based on text content.

a *Input*:
  - A document collection
  - $n$, the final number of desired clusters
  - $\sigma$, the internal similarity threshold

b *Step one, tag-based clustering*:
  - Based on tag-features only, perform $k$-means with $k = n$
  - Keep the $m$ clusters with an internal similarity higher than $\sigma$

c *Step two, text-based clustering*:
  - Based on text-features only, perform $k$-means with $k = n - m$

d *Finally*:
  - The $m$ tags-based clusters and the $(n - m)$ text-based clusters are combined to form the final $n$-clustering

**Fig. 2.** The 2-step approach: tags then text

In practice, this algorithm is as fast as a text-based $n$-clustering (often faster). This is due to the fact that tag-based clustering is very efficient thanks to a representation with a very small number of features.

**Integrating a new structural indicator: The T/E measure.** The T/E measure is a structural indicator of the proportion of "mixed content" in an XML fragment. In previous research, it has given us the Full-Text Likelihood of an XML element, based on which, the element could be excluded from a full-text index [14]. Although the values of the T/E measure are in the continuous range from 0 to $\infty$, the interpretation has come with a projection into a binary value space, where values greater than 1.0 provide evidence of full-text content. When treated as a feature for clustering XML documents, the projection is unnecessary. Therefore, the whole value space of the T/E measure is available. The T/E measure is a quite reliable indicator when the XML fragment is relatively small, e.g. a paragraph of text or a small section. As the size or the heterogeneity of the fragment increases, a single T/E value starts to shift from being an exact indicator towards being an approximation.

**Integrating the T/E measure into the vector space model** We integrate the T/E measure as an additional dimension of the vector space. Because our weighting scheme is based on inverted document frequency, the inclusion of the T/E value for every document would have a null effect. Therefore, we only gave a non-null value to the T/E dimension if the T/E measure was greater than 1.

## 4 Evaluation

In this section, we will define the experimental settings. We will first describe the document collections and the evaluation measures, and then present our results in details.

### 4.1 Collection description

The INEX mining track 2006 provides two separate XML document collections. The first one is a collection of scientific journal articles from the IEEE Computer Society[3]. The second one is a collection of English documents from Wikipedia [15]. Each collection further includes a set of categories $C$. Every document is assigned to a subset of categories of $C$ that describe it best. The goal of the clustering task is to automatically produce a categorization that matches these (*ideal*) assignments as closely as possible.

A specificity in the INEX mining track 2006 is that there is exactly one category corresponding to every document. In other words, each collection is partitioned into category-wise subcollections.

Let us now describe the two collections in further details.

**IEEE.** The IEEE collection has long been known as the "INEX collection", because it was the only collection in use in the main INEX track from the first INEX initiative in 2002 until 2006. It contains approximately $12,000$ articles published in 18 different IEEE journals. They are mainly marked up with hierarchical and stylistic elements. The hierarchical markup typically indicates the beginning and end of sections, subsections and paragraphs, and possibly their titles, as well as figures and bibliographical references. Stylistic elements, for instance, are used to mark bolded text or mathematical formulas.

The categories that were used to partition the collection are the journals in which a document was originally published. Hence, every document is assigned to exactly one category.

As we pointed out earlier [7], we believe that these categories are not fully satisfying, as the fact that a paper was published in a given journal does not necessarily mean that it is entirely irrelevant to every other journal. Among other things, such a strict interpretation means we should assume that a paper published in "Transactions on Computers" cannot possibly have anything in common with a paper published in "Transactions on Parallel&Distributed Systems".

Moreover, the IEEE collection contains documents of different types. The most common document type is scientific articles, but the collection also contains calls for papers, book reviews, keyword indices, etc. Regardless of their nature, documents published in the same journal are assigned to the same category. An intuitive issue with this "ideal" categorization is that any clustering assigning documents by their nature will be penalized in the evaluation process.

**Wikipedia.** The Wikipedia collection is new to INEX 2006. It contains $150,094$ English documents from Wikipedia. The collection used in the mining track is a subset of the "main" Wikipedia collection as described by Denoyer and Gallinari [15]. The main collection contains $659,388$ documents and covers a

[3] http://www.computer.org/

hierarchy of 113,483 categories. It contains about 5,000 different tags, with an average number of 161 XML nodes per document and an average element depth of 6.72.

The subset of the Wikipedia collection used in the INEX mining track consists of the 150,094 documents to which only one semantic category corresponds. These categories were extracted from the Wikipedia portals, which include 72 semantic categories (the 113,483 categories mentioned earlier come from a different source, check [15] for details). After the removal of documents to which more than 1 category was attached, only 60 non-empty classes remained. This partition is used as the evaluation gold-standard.

Naturally, we may express similar concerns as the ones we expressed earlier about the IEEE collection. The assumption that a document should belong to one and only one category does not seem right when we are handling text. To use a partition as our ideal classification implies the assumption that no two categories have anything in common. This can hardly be right when those categories are based on semantics.

### 4.2 Evaluation measures

As we mentioned earlier, the evaluation of the clustering track relies on the comparison of a given run to an ideal classification. The theoretical gold standards for each collection were described in the previous subsection. We shall now introduce the metrics of this comparison. In the INEX mining track, two official measures were used to compare an ideal classification and an experimental run: the micro- and macro-average F1 measures. We define these measures below.

**Recall and precision.** For a given category, we define the positives (respectively, negatives) as the set of documents assigned to that category (respectively, not assigned to that category).

When we compare a submission to the ideal classification, we define the true positives (TP) as the positives that were assigned to the right category. The false positives (FP) are the documents that were wrongly assigned to that category. Similarly, the true negatives (TN) were duly assigned to another category, while the false negatives (FN) should have been assigned to the category being considered.

Precision and recall are defined as follows:

$$Precision = \frac{TP}{TP + FN},\ Recall = \frac{TP}{TP + TN}.$$

**F1-measure.** Precision and recall complement each other. For instance, it is easy to obtain very high scores with one, at the expense of the other. To get perfect recall, one can simply assign every document to every category. In a symmetrical fashion, one may obtain high precision by limiting the number of

answers. Hence, we rather utilize a measure that combines precision and recall, such as the F1-measure, defined as the harmonic average of recall and precision:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

**Micro- and macro-average.** To obtain a single measure for the evaluation of a classification, the F1 measure needs to be averaged over all the classes. There are two ways to do this. *Macro-average F1* is the non-weighted average of the F1 measure over all the classes, while *micro-average F1* is weighted by the number of documents in each corresponding class. Clearly, the latter is more strongly influenced by larger classes.

### 4.3 Experimental results

We submitted the runs previously described to the INEX mining track 2006. The document features were weighted with inverted document frequency, and the Cluto software[4] was used to perform the $k$-means clusterings.

All the submissions, ours and those of other participants, returned the same number of categories as in the ideal classifications: 18 for the IEEE collection and 60 for the Wikipedia collection.

Our results are summarized in Table 1 for the IEEE collection and in Table 2 for the Wikipedia collection. The notation "Tags→Text, 0.8" means that we used the "tags then text" approach, and kept the tag-based clusters with an internal similarity higher than 0.8.

**Table 1.** INEX - IEEE Collection

| Features | Micro F1 | Macro F1 | Overall rank (out of 13) |
|---|---|---|---|
| Text | .348789 | .290407 | 3rd |
| Tags | .132453 | .081541 | 7th |
| Text+Tags | .253369 | .203533 | 5th |
| Tags→Text, 0.8 | .270350 | .222365 | 4th |
| Text + T/E | .348828 | .290379 | 2nd |

### 4.4 Result analysis

**Overall observations.** Looking at the results of our submissions, we can make a number of observations. An obvious disappointment is the fact that the exclusive use of text features beats all the other alternatives by far. Even worse,

[4] CLUTO, http://www-users.cs.umn.edu/∼karypis/cluto/

**Table 2.** INEX - Wikipedia Collection

| Features | Micro F1 | Macro F1 | Overall rank (out of 7) |
|---|---|---|---|
| Text | .444455 | .210621 | 1st |
| Tags | .221829 | .072834 | 6th |
| Text+Tags | .372376 | .128239 | 5th |
| Tags→Text, 0.8 | .406129 | .155034 | 4th |
| Tags→Text, 0.9 | .413439 | .159473 | 3rd |
| Text + T/E | .427438 | .183567 | 2nd |

when we look at "text", "tags", "text+tags" and "text+T/E", it seems like the performance decreases as the number of structural features increases.

On the positive side, we could confirm that tag-based clustering is very fast, and that using the "tags then text" approach performs just as fast as using text features only.

Another positive result is that "tags then text" outperforms "text+tags". This result is especially satisfying because both approaches use exactly the same features. Consequently, we get confirmation that the 'tags then text" technique is a better way to integrate structural features into the clustering process.

The explanation is fairly simple. The structural clustering can detect and put aside what we may call "structural outliers". Typically, in the IEEE collection, they are tables of contents and keyword indices of journal issues as well as calls for papers. In the Wikipedia collection, the outliers include lists (lists of counties by area, list of English cricket clubs, etc.). However, to count on a small number of element names as unique document descriptors is obviously risky. This is why we ensure that only the most cohesive tag-based clusters are kept, by using a high internal similarity threshold.

**Comparison with other participants.** In the INEX mining track, a total of 7 clustering runs were submitted for the Wikipedia collection and 13 for the IEEE collection.

*On the Wikipedia collection.* As shown in table 2, our 6 Wikipedia submissions rank at the first 6 places of the INEX clustering track 2006. Only one other team submitted a run for the Wikipedia collection. This is mostly due to scalability issues. Several approaches are indeed based on XML tree operations, which are computationally complex and may become intractable with a shift from $12,017$ documents (IEEE) to $150,094$ (Wikipedia), combined with the fact that the structure of the Wikipedia documents is much deeper and much more unpredictable (from 163 distinct elements to $7,208$). Another reason that might have discouraged potential participants is the lack of a DTD. This is, however, a very common feature of real-life collections.

One strong point in our approach is that it does not use anything but the documents themselves. From a computational point of view, clustering is the costliest operation with a linear time complexity of $O(n)$ or $O(n \log n)$.

Hence we had no problems shifting from one collection to the other and we do not expect difficulties in applying this work to new collections, whatever their structure and size is, since our techniques scala well.

*On the IEEE collection.* Two other teams submitted clustering runs for the IEEE collection. The overall ranking of our runs is given in table 1. The best-performing method is based on contextual self-organizing maps [9]. Its performance is fairly close to our own best, with a micro-average F1 of 0.365079 and a macro-average of 0.326469. The technique proves to be efficient. However, its complexity makes it hardly scalable (the supervised learning actually needs to be restricted to a subpart of the IEEE document trees: the content of the "fm" element).

*Conclusion.* We should be quite happy to see our runs in the top ranks for both collections. However, the fact that our best run is always the one that actually ignores the structural information is rather worrisome. We believe that this is not necessarily due to a weak state of the art of the systems presented in the INEX clustering task, but for a big part to a semantic bias of the evaluation system.

### 4.5 Discussion on evaluation

**Evaluation of clustering.** There are two ways to evaluate clustering experiments. The first one is to use *internal quality measures*, such as entropy, purity, or cohesiveness. For instance, the cohesiveness of a cluster is the average similarity between each two documents in the cluster. The problem of these measures is that the computation of document similarities is strongly dependent on the document model. Internal quality measures are useful to compare clustering techniques based upon the same document model, but they are meaningless in most other cases. In particular, they cannot help as we wish to compare techniques based on the same algorithm but different feature sets.

In such situations, we must rely on *external quality measures*, such as recall, precision, or F1-measure. The latter were the official evaluation metrics for the clustering task of the INEX mining track in 2006.

**Gold-standard.** External measures are meant to compare every submitted clustering to a "gold-standard" classification. The more similar a run is to that standard, the better. Defining such a gold-standard is a great challenge.

Indeed, we are not convinced that the gold-standard classifications that were used for the evaluation of the INEX mining track are optimal. The consequence of this is very important, because to improve a system's performance with respect to the F1-measure means to produce a classification closer to the gold-standard. If the gold-standard is weak, improving the performance of a system might actually require that a number of reasonable assumptions be compromised.

**What is a good clustering?** The main issue with the current "gold" classifications is the use of disjoint clusters. This is an excessive simplification when we are dealing with text and thematic classes, as is the case currently.

In fact, having to deal with thematic classes can also be seen as a problem. Since the motivation of XML clustering is to take structural information into account, we should also consider categories that are not solely based on semantics.

An empirical analysis of our clusters show that the technique "tags then text" manages to put aside outliers, such as tables of contents or call for papers in the IEEE collection. Our technique stores these into clusters of their own and performs text-based clustering with the remaining documents. We do believe that this is a good result for most uses of the document collection. Thinking of information retrieval, it is likely that a user performing a search on a scientific journal is looking for articles (or fragments thereof) rather than keyword indices or calls for papers. However, with respect to the current evaluation metrics, the effectiveness of a system taking this fact into account is weakened, because the gold-standards were built the opposite way: each call for papers belongs to the journal in which it was published. Hence, they are spread out uniformly in the ideal classification, while we actually built a system that puts all of them in the same category.

The problem is that if the gold-standards were solely based on the document structure, separating calls for papers, tables of contents and regular articles, one would as well be able to argue that it does not make sense to have to categorize the call for papers for a data mining conference is in a different class from that of a data mining article. The key issue is there. Given a document collection, there are numerous "perfect" ways to classify the contents. The classification needs to be related to a certain need, but it may still be totally inappropriate in other situations.

This leads us to the conclusion that the intended use of XML document clustering needs to play a larger role in its evaluation, and hence a prior question needs to be answered: why do we do it? If the goal of XML clustering is to build a semantic-based disjoint taxonomy, then the current gold-standards are suitable. In order to detect DTDs automatically, we would need a structure-oriented gold standard. For information retrieval, we might use the per topic relevance judgements as the classes, or perform indirect evaluation through the ad hoc XML IR runs.

## 5 Conclusion - Future work

Our conclusive remarks and suggestions concern two topics, our XML clustering approach and the general problem of the evaluation of XML clustering. Actually, the evaluation of supervised classification is also related, even though the learning phase can help compensate for the issues aforementioned.

We have introduced the experiments with our XML clustering techniques in the context of the INEX 2006 mining track. The generality and scalability of our

approach was underlined by the fact that we made no difference in the way we handled two radically different document collections, whereas many participants have been discouraged by the size and depth of the Wikipedia document collection (perhaps also by the lack of a DTD). One weakness of our techniques is their flat use of the structural information. We created a "bag of structure" and implemented advanced ways to use it as a complement of the "bag of words", but we ignored the tree structure of the elements and did not either connect the words to their path in the XML tree. This is left for future work.

For both collections, we had the satisfaction to see our runs in the top ranks. Looking at the top 5 runs for the two collections, the only one that is not ours occupies the 1st rank for the IEEE collection. The "tags then text" approach was demonstrated to be more efficient at combining semantics and structure, than a baseline merger of the features, in spite of a tendency to contradict some of the arguable implications of the current evaluation system. Hence, in a more appropriate evaluation setting, we expect to observe the same phenomenon with an even greater margin. Evidently, this remains to be verified.

A source of concern should be the fact that our best performing runs were the ones that actually ignored the structural information. However, we feel that this only reflects the bias of the evaluation system. Indeed, micro- and macro-average F1 are measuring the closeness of a run to a theoretically ideal classification. However, the current "ideal" classifications in use are disjoint and thematic. Since there is no evidence that the classifications we use as gold-standards are related to the structure of the documents, it is natural that the best performing approaches are the ones that simply ignore that structure.

An important point is that several classifications of the same collection may be perfect, depending on the context. We hence plead for placing the applications of XML clustering in the center of the evaluation process. This may be done by creating an ideal classification for every corresponding application, and/or by evaluating XML clustering indirectly, by measuring how much we can benefit from it in another task. In 2006, an INEX "user case studies track" was created. Perhaps a comparable reflection is now needed in the XML mining track.

## References


1. Jardine, N., van Rijsbergen, C.: The use of hierarchic clustering in information retrieval. Information Storage and Retrieval **7** (1971) 217–240
2. Tombros, A.: The effectiveness of hierarchic query-based clustering of documents for information retrieval. PhD thesis, University of Glasgow (2002)
3. Guillaume, D., Murtaugh, F.: Clustering of XML Documents. Computer Physics Communications **127** (2000) 215–227
4. Yi, J., Sundaresan, N.: A classifier for semi-structured documents. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts. (2000) 340–344
5. Nierman, A., Jagadish, H.: Evaluating Structural Similarity in XML. In: Fifth International Workshop on the Web and Databases (WebDB 2002), Madison, Wisconsin. (2002)

6. Denoyer, L., Gallinari, P.: Report on the xml mining track at inex 2005 and inex 2006. [16]
7. Doucet, A., Ahonen-Myka, H.: Naive clustering of a large xml document collection. In: Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Schloss Dagsuhl, Germany (2002) 81–87
8. Yong, S.L., Hagenbuchner, M., Tsoi, A., Scarselli, F., Gori, M.: Xml document mining using graph neural network. In Fuhr, N., Lalmas, M., Malik, S., Kazai, G., eds.: INEX. Volume 3977 of Lecture Notes in Computer Science., Springer (2006)
9. Kc, M., Hagenbuchner, M., Tsoi, A.C., Scarselli, F., Gori, M., Sperduti, A.: Xml document mining using contextual self-organizing maps for structures. [16]
10. Despeyroux, T., Lechevallier, Y., Trousse, B., Vercoustre, A.M.: Experiments in clustering homogeneous xml documents to validate an existing typology. In: Proceedings of the 5th International Conference on Knowledge Management (I-Know), Vienna, Austria, Journal of Universal Computer Science (2005)
11. Vercoustre, A.M., FEGAS, M., Lechevallier, Y., Despeyroux, T.: Classification de documents xml à partir d'une représentation linéaire des arbres de ces documents. In: Actes des 6èmes journées Extraction et Gestion des Connaissances (EGC 2006), Revue des Nouvelles Technologies de l'Information (RNTI-E-6), Lille, France (2006)
12. Candillier, L., Tellier, I., Torre, F.: Transforming xml trees for efficient classification and clustering. INEX 2005 Workshop on Mining XML documents (2005)
13. Willett, P.: Recent trends in hierarchic document clustering: a critical review. In Information Processing and Management **24** (1988) 577–597
14. Lehtonen, M.: Preparing Heterogeneous XML for Full-Text Search. ACM Transactions on Information Systems **24** (2006) 1–21
15. Denoyer, L., Gallinari, P.: The Wikipedia XML Corpus. SIGIR Forum (2006)
16. Fuhr, N., Lalmas, M., Malik, S., Kazai, G., eds.: Advances in XML Information Retrieval and Evaluation, 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 18-20 2006, Revised Selected Papers. In Fuhr, N., Lalmas, M., Malik, S., Kazai, G., eds.: INEX. Lecture Notes in Computer Science, Springer (2007)