Overview of the INEX 2009 XML Mining Track: Clustering and Classification of XML Documents

Richi Nayak Christopher M. De Vries Sangeetha Kutty Shlomo Geva

Faculty of Science and Technology
Queensland University of Technology
GPO Box 2434, Brisbane Qld 4001, Australia

{r.nayak, christopher.devries, s.kutty, s.geva}@qut.edu.au

Ludovic Denoyer Patrick Gallinari

University Pierre et Marie Curie
LIP6 – 104 avenue du président Kennedy
75016 PARIS - FRANCE

{Ludovic.denoyer, Patrick.gallinari}@lip6.fr

Abstract. This report explains the objectives, datasets and evaluation criteria of both the clustering and classification tasks set in the INEX 2009 XML Mining track. The report also describes the approaches and results obtained by the different participants.

Keywords: XML document mining, INEX, Wikipedia, Structure and content, Clustering, Classification.

1 Introduction

The XML Document Mining track was launched for exploring two main ideas: (1) identifying key problems and new challenges of the emerging field of mining semi-structured documents, and (2) studying and assessing the potential of Machine Learning (ML) techniques for dealing with generic ML tasks in the structured domain i.e. classification and clustering of semi structured documents. This track has run for five editions during INEX 2005, 2006, 2007, 2008 and 2009. The four first editions have been summarized in [2, 3, 4] and we focus here on the 2009 edition.

INEX 2009 included two tasks in the XML Mining track: (1) unsupervised clustering task and (2) semi-supervised classification task where documents are organized in a graph. The clustering task requires the participants to group the documents into clusters without any knowledge of cluster labels using an unsupervised learning algorithm. On the other hand, the classification task requires the participants to label

the documents in the dataset into known classes using a supervised learning algorithm and a training set. This report gives the details of clustering and classifications tasks.

2 The Clustering Track

In the last decade, we have observed a proliferation of approaches for clustering XML documents based on their structure and content [9,12]. There have been many approaches developed for diverse application domains. Many applications require data objects to be grouped by similarity of content, tags, paths, structure and semantics. The clustering task in INEX 2009 evaluates clustering approaches in the context of XML information retrieval.

The INEX 2009 clustering task is different from the previous years due to its incorporation of a different evaluation strategy. The clustering task explicitly tests the Jardine and van Rijsbergen cluster hypothesis (1971) [8], which states that documents that cluster together have a similar relevance to a given query. It uses manual query assessments from the INEX 2009 Ad Hoc track. If the cluster hypothesis holds true, and if suitable clustering can be achieved, then a clustering solution will minimise the number of clusters that need to be searched to satisfy any given query. There are important practical reasons for performing collection selection on a very large corpus. If only a small fraction of clusters (hence documents) need to be searched, then the throughput of an information retrieval system will be greatly improved. INEX 2009 clustering task provides an evaluation forum to measure the performance of clustering methods for collection selection on a huge scale test collection. The collection consists of a set of documents, their labels, a set of information needs (queries), and the answers to those information needs.

2.1 Corpus

The INEX XML Wikipedia collection is used as a dataset in this task. This 60 Gigabyte collection contains 2.7 million English Wikipedia XML documents. The XML mark-up includes explicit tagging of named entities and document structure. In order to enable participation with minimal overheads in data-preparation the collection was pre-processed to provide various representations of the documents. For instance, a bag-of-words representation of terms and frequent phrases in a document, frequencies of various XML structures in the form of trees, links, named entities, etc. These various collection representations made this task a lightweight task that required the participants to submit clustering solutions without worrying about pre-processing this huge data collection.

There are a total of 1,970,515 terms after stemming, stopping, and eliminating terms that occur in a single document for this collection. There are 1,900,075 unique terms that appear more than once enclosed in entity tags. There are 5213 unique entity tags in the collection. There are a total of 110,766,016 unique links in the collection. There

are a total of 348,552 categories that contain all documents except for a 118,685 document subset containing no category information. These categories are derived by using the YAGO ontology [16]. The YAGO categories appear to follow a power law distribution as shown in Figure 1. Distribution of documents in the top-10 cluster category is shown in Table 1.

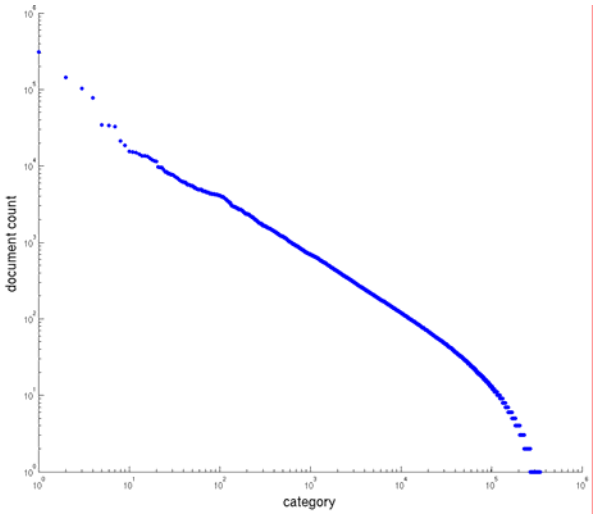


Figure 1: The YAGO Category Distribution

Category	**Documents**
Living people	307304
All disambiguation pages	143463
Articles with invalid date parameter in template	77659
All orphaned articles	34612
All articles to be expanded	33810
Year of birth missing (living people)	32499
All articles lacking sources	21084
Human name disambiguation pages	18652
United States articles missing geocoordinate data	15363
IUCN Red List least concern species	15241

Table 1: Top-10 Category Distribution

A subset of collection containing about 50,000 documents (of the entire INEX 2009 corpus) was also used in the task to evaluate the categories labels results only, for teams that were unable to process such a large data collection.

2.2 Tasks and Evaluation Measures

The task was to utilize unsupervised classification techniques to group the documents into clusters. Participants were asked to submit multiple clustering solutions containing different numbers of clusters such as 100, 500, 1000, 2500, 5000 and 10000. The clustering solutions are evaluated by two means. Firstly, we utilise the *classes-to-clusters evaluation* which assumes that the classification of the documents in a sample is known (i.e., each document has a class label). Then any clustering of these documents can be evaluated with respect to this predefined classification. It is important to note that the class labels are not used in the process of clustering, but only for the purpose of evaluation of the clustering results.

The standard criterion of purity is used to determine the quality of clusters. These evaluation results were provided online and ongoing, starting from mid-October. Entropy and F-Score were not used in evaluation. The reason behind was that a document in the corpus maps to more than one category. Due to multi labels that a document can have, it was possible to obtain higher value of Entropy and F-Score than the ideal solution. Purity measures the extent to which each cluster contains documents primarily from one class. Each cluster is assigned with the class label of the majority of documents in it. The macro and micro purity of a clustering solution *cs* is obtained as a weighted sum of the individual cluster purity. In general, larger the value of purity, better the clustering solution is.

$$\text{Purity (k)} = \frac{\text{Number of documents with the majority label in cluster k}}{\text{Numberof documents in cluster k}}$$

$$\text{Micro-Purity (cs)} = \frac{\sum_{k=0}^{n} \text{Purity(k)} * \text{TotalFoundByClass(k)}}{\sum_{k=0}^{n} \text{TotalFoundByClass(k)}}$$

$$\text{Macro-Purity (cs)} = \frac{\sum_{k=0}^{n} \text{Purity(k)}}{\text{Total Number of Categories}}$$

The clustering solutions are also evaluated to determine the quality of cluster relative to the optimal collection selection goal, given a set of queries. Better clustering solutions in this context will tend to (on average) group together relevant results for (previously unseen) ad-hoc queries. Real Ad-hoc retrieval queries and their manual assessment results are utilised in this evaluation. This novel approach evaluates the clustering solutions relative to a very specific objective - clustering a large document collection in an optimal manner in order to satisfy queries while minimising the search space. The Normalised Cumulative Gain is used to calculate the score of the

best possible collection selection according to a given clustering solution of *n* number of clusters. Better the score when the query result set contains more cohesive clusters. The cumulative gain of a cluster (CCG) is calculated by counting the number of relevant documents in a cluster, *c*, for a topic, *t*, where *c* is the set of documents in a cluster and *t* is the set of relevant documents for a topic.

$$CCG(c,t) = |c \cap t|$$

For a clustering solution for a given topic, a (sorted) vector CG is created representing each cluster by its CCG value. Clusters containing no relevant documents are represented by a value of zero. The cumulated gain for the vector CG is calculated which is then normalized on the ideal gain vector. Each clustering solution *cs* is scored for how well it has split the relevant set into clusters using *CCG* for the topic *t*.

$$SplitScore(t,cs) = \frac{\sum^{|CG|} \text{cumsum(CG)}}{\text{nr}^2}$$

nr = Number of relevant documents in the returned result set for the topic t.

A worst possible split is assumed to place each relevant document in a distinct cluster. Let CG1 be a vector that contains the cumulative gain of every cluster with a document each.

$$MinSplitScore(t,cs) = \frac{\sum^{|CG1|} \text{cumsum(CG1)}}{\text{nr}^2}$$

The normalized cluster cumulative gain (nCCG) for a given topic *t* and a clustering solution *cs* is given by,

$$\text{nCCG}(t,cs) = \frac{SplitScore(t,cs) - MinSplitScore(t,cs)}{1 - \text{Min}SplitScore(t,cs)}$$

The mean and the standard deviation of the nCCG score over all the topics for a clustering solution *cs* are then calculated. n is total number of topics.

$$\text{Mean nCCG(cs)} = \frac{\sum_{t=0}^{n} \text{nCCG}(t,cs)}{\text{n}}$$

$$\text{Std Dev nCCG(cs)} = \frac{\sum_{t=0}^{n} (\text{nCCG}(t,cs) - \text{Mean nCCG(cs)})^2}{\text{n}}$$

A total of 68 topics were used to evaluate the quality of clusters generated on the full set of collection of about 2.7 million documents. A total of 52 topics were used to evaluate the quality of clusters generated on the subset of collection of about 50,000

documents. A total number of 4858 documents were found relevant by the manual assessors for the 68 topics. An average number of 71 documents were found relevant for a given topic by manual assessors. The nCCG value varies from 0 to 1.

2.3 Participants, Submissions and Evaluation

A total of six research teams have participated in the INEX 2009 clustering task. Two of them submitted the results for the subset data only. We briefly summarised the approaches employed by the participants.

Exploiting Index Pruning Methods for Clustering XML Collections [1]
[1] used Cover-Coefficient Based Clustering Methodology (C3M) to cluster the XML documents. C3M is a single-pass partitioning type clustering algorithm which measures the probability of selecting a document given a term that has been selected from another document. As another approach, [1] adapted term-centric and document-centric index pruning techniques to obtain more compact representations of the documents. Documents are clustered with these reduced representations for various pruning levels, again using C3M algorithm. All of the experiments are executed on the subset of INEX 2009 corpus including 50K documents.

Clustering with Random Indexing K-tree and XML Structure [5]
The Random Indexing (RI) K-tree has been used to cluster the entire 2,666,190 XML documents in the INEX 2009 Wikipedia collection. Clusters were created as close as possible to the 100, 500, 1000, 5000 and 10000 clusters required for evaluation. The algorithm produces clusters of many sizes in a single pass. The desired clustering granularity is selected by choosing a particular level in the tree. In the context of document representation, topology preserving dimensionality reduction is preserving document meaning – or at least this is the conjecture which the team tests here Document structure has been represented by using a bag of words and a bag of tags representation derived from the semantic markup in the INEX 2009 collection. The term frequencies were weighted with BM25 where K1 = 2 and b = 0.75. The tag frequencies were not weighted.

Exploiting Semantic tags in XML Clustering [10]
This technique combines the structure and content of XML documents for clustering. Each XML document in the INEX Wikipedia corpus is parsed and modeled as a rooted labeled ordered *document tree*. A constrained frequent subtree mining algorithm is then applied to extract the common structural features from these document trees in the corpus. Using the common structural features, the corresponding content features of the XML documents are extracted and represented in a Vector Space Model (VSM). The term frequencies in the VSM model were weighted with both TF-IDF and BM25. There were 100, 500 and 1000 clusters created for evaluation.

Performance of K-Star at the INEX'09 Clustering Task [13]
The employed approach was quite simple and focused on high scalability. The team used a modified version of the Star clustering method which automatically obtains the number of clusters. In each iteration, this clustering method brings together all those items whose similarity value is higher than a given threshold T, which is typically assumed to be the similarity average of the whole document collection and, therefore, the clustering method "discover" the number of clusters by its own. The run submitted to the INEX clustering task split the complete document collection into small subsets which are clustered with the above mentioned clustering method.

Evaluation

Figure 2, Figure 3 and Figure 4 show the performance of various teams in the clustering task. The legends are formatted in the following fashion, [metric] – [institution] (username) [method].

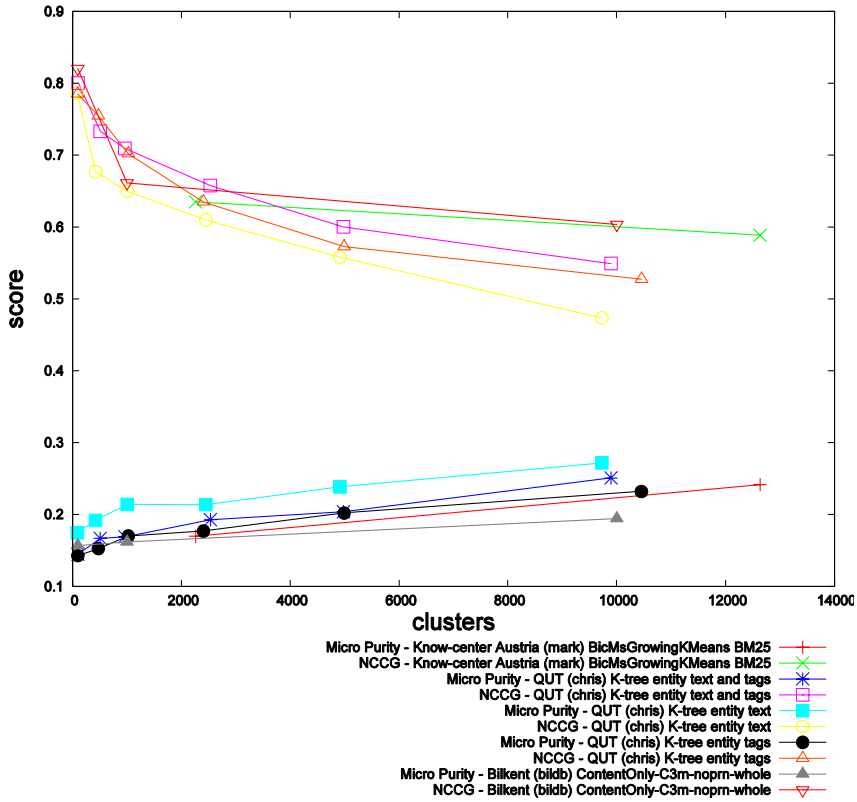


Figure 2: Purity and NCCG performance of different teams using the entire dataset

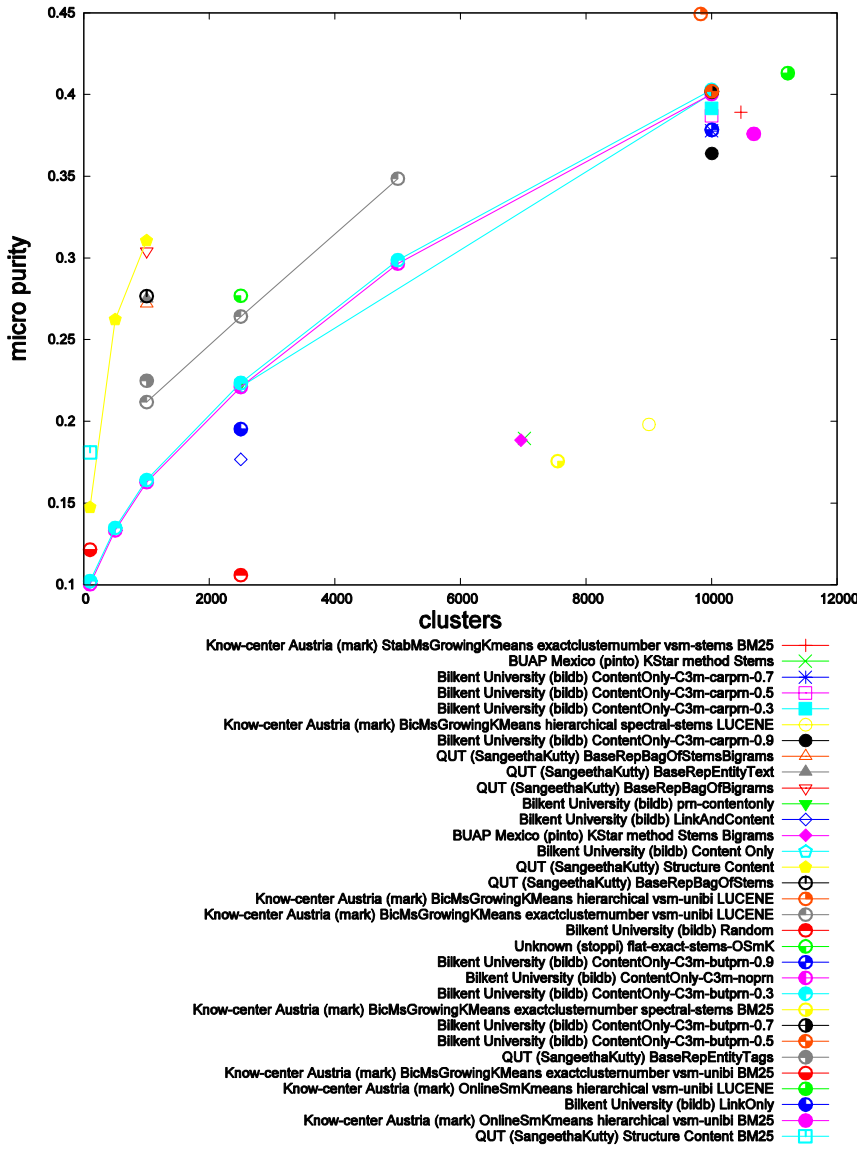


Figure 3: Purity performance of different teams using the subset data

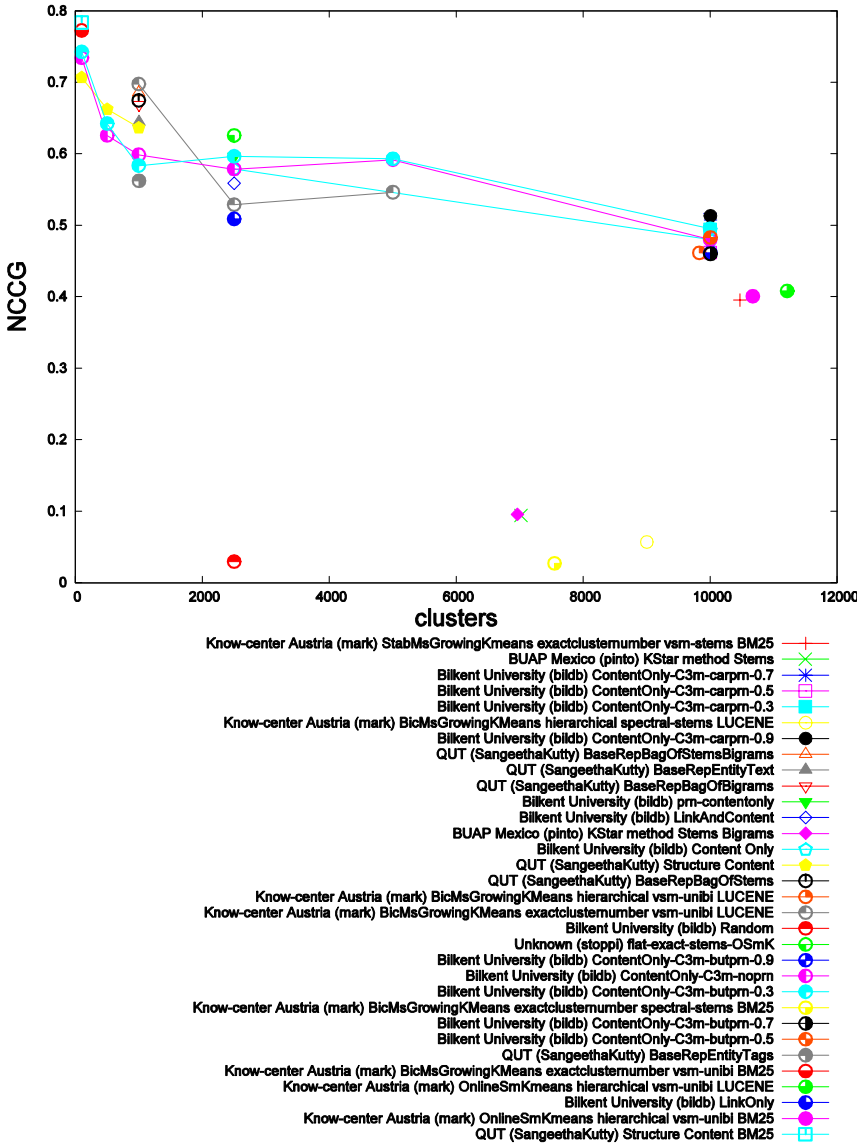
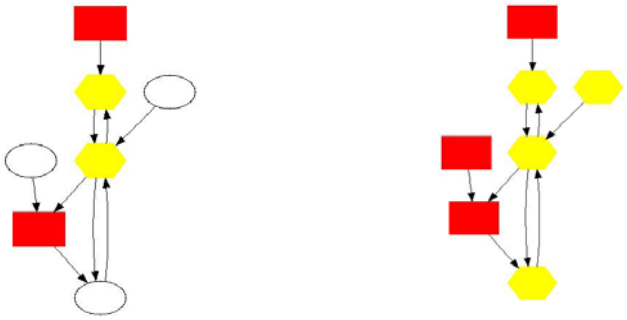


Figure 4: NCCG performance of different teams using the subset data

3. The Classification Track

Dealing with XML document collections is a particularly challenging task for ML and IR. XML documents are defined by their logical structure and their content (hence the name semi-structured data). Moreover, in a large majority of cases (Web collections for example), XML documents collections are also structured by links between documents (hyperlinks for example). These links can be of different types and correspond to different information: for example, one collection can provide hierarchical links, hyperlinks, citations, ... Most models developed in the field of XML categorization simultaneously use the content information and the internal structure of XML documents (see [2] and [3] for a list of models) but they rarely use the external structure of the collection i.e the links between documents. Some methods using both content and links have been proposed in [4].

The XML Classification Task focuses on the problem of learning to classify documents organized in a graph of documents. Unlike the 2008 track, we consider here the problem of *Multiple labels classification* where a document belongs to one or many different categories. This task considers a tansductive context where, during the training phase, the whole graph of documents is known but the labels of only a part of them are given to the participants (Figure).



Training set Final labelling

Figure 5: The supervised classification task. Colors/Shapes correspond to categories, circle/white nodes are unlabeled nodes. Note that in this track, documents may belong to many categories

3.1 Corpus

The corpus provided is a subset of the *INEX 2009 Corpus*. We have extracted a set of 54,889 documents and the links between these documents. These links corresponds to the links provided by the authors of the Wikipedia articles. The documents have been transformed into TF-IDF vectors by the organizers. The corpus thus corresponds to a set of 54,889 vectors of dimension 186,723. The documents belong to 39 categories that correspond to 39 Wikipedia portals. We have provided the labels of 20 % of the documents. The corpus is composed of 4,554,203 directed links that correspond to hyperlinks between the documents of the corpus. Each document is concerned by 84.1 links on average.

Number of documents	54,889
Number of training documents	11,028
Number of test documents	43,861
Number of categories	39
Number of links	4,554,203
Number of distinct words	186,723

3.2 Evaluation Measures

In order to evaluate the submissions of the participants, we have used different measures. The first set of measures are computed over each category and then averaged over the categories (using a micro or a macro average):

- *Accuracy* (ACC) corresponds to the classification error. Note that a system that returns zero relevant category for each document has a quite good accuracy.
- *F1 score* (F1) corresponds to the classical F1 measure and measures the ability of a system to find the relevant categories.

The second set of measures are computed over each document and then averaged over the documents.

- *Average precision* (APR) corresponds to the Average Precision computed over the list of categories returned for each document. It measures the ability of a

system to rank correctly the relevant categories. This measure is based on a ranking score of each category for each document.

3.3 Participants and Submissions

Five different teams have participated to the track. They have submitted different runs and we present here only the best results obtained by each team. Note that, due to additional experiments made after the submission deadline, the results presented here and the results presented in the participants' articles can be different.

Team	Micro ACC	Macro ACC	Micro F1	Macro F1	APR
University of Wollongong	92.5	94.6	51.2	47.9	68
University of Peking	94.7	96.2	51.8	48	70.2
XEROX Research Center	96.3	97.4	**60**	**57.1**	67.8
University of Saint Etienne	96.2	97.4	56.4	53	68.5
University of Granada	67.8	75.4	26.2	25.3	**72.9**

3.4 Summary of the methods

We give here a brief description of the methods submitted by the participants. Please refer to the participants articles for a detailed description of the methods and for the final results obtained by the different teams.

Multi-label Wikipedia classification with textual and graph features [6]

This paper proposes to evaluate different classification methods used on both the textual features of the pages to classify, and also on graph features computed from the structure of the graph. These features include for example the mean centrality, the degree centrality, etc...Different classifiers have been tested to handle the multi-label problem.

Supervised Encoding of Graph-of-Graphs for Classification and Regression Problems [7]

This article proposes a novel method which aims at encoding graph of graph structures where data correspond to a graph of elements which are also composed of graphs. The graph to graph structure is described and then used as a classification model based on a back-propagation of the error through the different level of the nested structure.

UJM at INEX 2009 XML Mining Track [11]

The authors use different classification strategies based on a set of content features to handle the classification problem. They mainly compare different features selection methods and thresholding strategies.

Link-based text classification using Bayesian networks [14]

The article presents a Bayesian network model that is able to handle both content and links between documents. The proposed model is an extension of the Naïve Bayes model to documents organized in a graph.

Extended VSM for XML Document Classification using Frequent Subtrees [15]

The last paper proposes the structured link vector model which aims at modeling both the content and the structure of the documents in a vector. Mainly, the authors propose to insert into classical content-based features vectors information about the frequent XML subtrees and the links between documents

4. Conclusion

The XML Mining track in INEX 2009 brought together researchers from Information Retrieval, Data Mining, Machine Learning and XML fields. The clustering task allowed participants to evaluate clustering methods against a real use case and with significant volumes of data. The task was designed to facilitate participation with minimal effort by providing not only raw data, but also pre-processed data which can be easily used by existing clustering software. The classification task allowed participant to explore algorithmic, theoretical and practical issues regarding the classification of interdependent XML documents.

5. Acknowledgments

We would like to thank all the participants for their efforts and hard work.

6. References

1. Altingovde I, Atilgan D, Ulusoy O., Exploiting Index Pruning Methods for Clustering XML Collections. In INEX 2009 Preproceedings.
2. Denoyer, L., Gallinari, P.: Report on the xml mining track at inex 2005 and inex 2006: categorization and clustering of xml documents. 41(1) (2007) 79–90
3. Denoyer, L., Gallinari, P.: Report on the xml mining track at inex 2007 categorization and clustering of xml documents. 42(1) (2008) 22–28
4. Denoyer, L., Gallinari, P.: Overview of the inex 2008 xml mining track. In: INEX. (2008) 401–411
5. De Vries C, Geva S, De Vine, L., Clustering with Random Indexing K-tree and XML Structure. In INEX 2009 Preproceedings.
6. Chidlovskii B.: Multi-label Wikipedia classification with textual and graph features. In: INEX. (2009)
7. Hagenbuchner M, Zhang S, Scarselli F, Chung Tsoi A.: Supervised Encoding of Graph-of-Graphs for Classification and Regression Problems. In: INEX. (2009)
8. Jardine, N. and van Rijsbergen, C. J., (1971) The Use of Hierarchic Clustering in Information Retrieval. Inform. Stor. ~ Retr., 7, 217- 240.
9. Kutty S, Nayak R, Li Y.: HCX: An Efficient Hybrid Clustering Approach for XML Documents. Proceedings of the ACM Document Engineering Symposium, Munich, Germany. (2009) 94-97
10. Kutty, S., Nayak, R., Li Y., Clustering XML documents using Multi-feature Model. In INEX 2009 Preproceedings.
11. Largeron C., Moulin C., Gery M., UJM at INEX 2009 XML Mining Track.In: INEX. (2009)
12. Nayak R., XML Data Mining: Process and Applications", Chapter 15 in "Handbook of Research on Text and Web Mining Technologies", Ed: Min Song and Yi-Fang Wu. Pp 249 -272, Publisher: Idea Group Inc., USA
13. Pinto, D, Tovar, M., Vilari˜no , D, Beltran, B., Salazar, H. BUAP: Performance of K-Star at the INEX'09 Clustering Task . In INEX 2009 Preproceedings.
14. E. Romero A, M de Campos L, Fernandez-Luna J. M., Huete J. F., Masegosa A. R.: Link-based text calssification using Bayesian networks. In: INEX (2009)
15. Yang J., Wang S.: Extended VSM for XML Document Classification using Frequent Subtrees. In: INEX. (2009)
16. Suchanek F, Kasneci, G, Weikum, G, YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In:WWW 2007.