UJM at INEX 2008 XML mining Track

Mathias Géry, Christine Largeron, Christophe Moulin

▶ **To cite this version:**

Mathias Géry, Christine Largeron, Christophe Moulin. UJM at INEX 2008 XML mining Track. Workshop INtitiative for Evaluation of XML Retrieval (INEX 2008), Jan 2008, Dagstuhl, Germany. pp.446-452. ujm-00366436

HAL Id: ujm-00366436
https://ujm.hal.science/ujm-00366436

Submitted on 26 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UJM at INEX 2008 XML mining track

Mathias Géry, Christine Largeron and Christophe Moulin

Université de Lyon, F-42023, Saint-Étienne, France
CNRS UMR 5516, Laboratoire Hubert Curien
Université de Saint-Étienne Jean Monnet, F-42023, France
{mathias.gery, christine.largeron, christophe.moulin}@univ-st-etienne.fr

Abstract. This paper[1] reports our experiments carried out for the INEX XML Mining track, consisting in developing categorization (or classification) and clustering methods for XML documents. We represent XML documents as vectors of indexed terms. For our first participation, the purpose of our experiments is twofold: Firstly, our overall aim is to set up a categorization text only approach that can be used as a baseline for further work which will take into account the structure of the XML documents. Secondly, our goal is to define two criteria (CC and CCE) based on terms distribution for reducing the size of the index. Results of our baseline are good and using our two criteria, we improve these results while we slightly reduce the index term. The results are slightly worse when we sharply reduce the size of the index of terms.

1 Introduction

The INEX XML Mining Track is organized in order to identify and design machine learning algorithms suited for XML documents mining [1]. Two tasks are proposed: clustering and categorization. Clustering is an unsupervised process through which all the documents must be classified into clusters. The problem is to find meaningful clusters without any prior information. Categorization (or classification) is a supervised task for which, given a set of categories, a training set of preclassified documents is provided. Using this training set, the task consists in learning the classes descriptions in order to be able to classify a new document in one of the categories.

This second task is considered in this article. Moreover, even if the content information (the text of the documents), the structural information (the XML structure of the documents) and the links between the documents can be used for this task, we have only exploited the textual information. Indeed, this is our first participation to this track and our aim was to design a framework that could be used as a baseline for further works dealing with structured documents.

More precisely, we focus on the preprocessing step, particularly the features selection, which is an usual step of the knowledge discovery process [8, 3, 2]. On

[1] This work has been partly funded by the Web Intelligence project (région Rhône-Alpes, cf. http://www.web-intelligence-rhone-alpes.org).

textual data, this step can be essential for improving the performance of the categorization algorithm. It exists a lot of words in the natural language, including stop words, synonymous, *etc.*. These words are not equally useful for categorization. Moreover, their distribution must also be considered. For example, words that appear in a single document are not useful for the categorization task.

So, we need to extract from the text a subset of terms that can be used to efficiently represent the documents in view of their categorization. In this paper, the documents are represented according to the vector space model (VSM [5]). Our aim is to adapt some VSM principles, for example the measure of the discriminatory power of a term, to the categorization task. We propose two criteria based on terms distribution aiming at extracting the indexing terms from the training set corpora. After a brief presentation of the VSM given to introduce our notations in section 2, these criteria are defined in the following section. Our categorization approach is described in section 3 while the experiments and the obtained results are detailed in sections 4 and 5.

2 Document model for categorization

2.1 Vector space model (VSM)

Vector space model, introduced by Salton and al. [5], has been widely used for representing text documents as vectors which contain terms weights. Given a collection D of documents, an index $T = \{t_1, t_2, ..., t_{|T|}\}$, where $|T|$ denotes the cardinal of T, gives the list of terms (or features) encountered in the documents of D. A document d_i of D is represented by a vector $\boldsymbol{d_i} = (w_{i,1}, w_{i,2}, ..., w_{i,|T|})$ where $w_{i,j}$ represents the weight of the term t_j in the document d_i. In order to calculate this weight, TF.IDF formula can be used.

2.2 TF: term representativeness

TF (Term Frequency), the relative frequency of term t_j in a document d_i, is defined by:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_l n_{i,l}}$$

where $n_{i,j}$ is the number of occurrences of t_j in document d_i normalized by the number of terms in document d_i. The more frequent the term t_j in document d_i, the higher is the $tf_{i,j}$.

2.3 IDF: discriminatory power of a term

IDF (Inverse Document Frequency) measures the discriminatory power of the term t_j. It is defined by:

$$idf_j = \log \frac{|D|}{|\{d_i : t_j \in d_i\}|}$$

where $|D|$ is the total number of documents in the corpus and $|\{d_i : t_j \in d_i\}|$ is the number of documents in which the term t_j occurs at least one time. The less frequent the term t_j in the collection of documents, the higher is the idf_j.

The weight $w_{i,j}$ of a term t_j in a document d_i is then obtained by combining the two previous criteria:

$$w_{i,j} = tf_{i,j} \times idf_j$$

The more frequent the term t_j is in document d_i and the less frequent it is in the other documents, the higher is the weight $w_{i,j}$.

3 Criteria for features selection

This VSM is widely used for text mining and information retrieval, as well for free format document like scientific articles as for semi structured document written in markup languages like HTML or XML.

But, in the context of categorization, even for limited collections, the dimensionality of the index can be exceedingly large. For example, in INEX collection, 652 876 non trivial words have been identified. This is a real problem for categorization since the terms belonging to this bag of words are not necessarily discriminatory features of the categories. So, we introduced two criteria (CC and CCE) in order to select a subset of T providing a description of the documents belonging to the same category. We consider that these terms must be very frequent in the documents of the category and, on the contrary, that they must be infrequent in the other categories.

3.1 Category Coverage criteria (CC)

Let df_j^k be the number of documents in the category C_k where term t_j appears, and f_j^k be the frequency of documents belonging to C_k and including t_j:

$$df_j^k = |\{d_i \in C_k : t_j \in d_i\}|, k \in \{1, ...r\} \quad (1)$$

$$f_j^k = \frac{df_j^k}{|C_k|} \quad (2)$$

The higher the number of documents of C_k containing t_j, the higher is f_j^k.

On the other hand, the term t_j could be considered as a discriminant term if most of the documents, where t_j appears, belongs to the same category. Thus, a first criteria, noted CC (Category Coverage), is computed as follows:

$$CC_j^k = \frac{df_j^k}{|C_k|} * \frac{f_j^k}{\sum_k f_j^k}$$

$$CC_j^k = \frac{(f_j^k)^2}{\sum_k f_j^k}$$

If the value of CC_j^k is high, then t_j is a characteristic feature of the category C_k.

3.2 Category Coverage Entropy criteria (CCE)

The frequency f_j^k considers the number of documents containing t_j but it does not take into account the number of occurrences of t_j in the category. It is the reason why we consider also p_j^k the frequency of t_j in the category C_k and a measure commonly used in information theory, called entropy, which evaluates the purity of the categories for the term t_j. In the context of text categorization, it measures the discriminatory power of t_j. Let n_j^k be the number of occurrences of t_j in the documents of C_k and p_j^k the corresponding frequency:

$$n_j^k = \sum_{d_i \in C_k} n_{i,j} \qquad\qquad p_j^k = \frac{n_j^k}{\sum_{k=1,r} n_j^k}$$

The Shannon entropy E_j of the term t_j is given by [6]:

$$E_j = -\sum_{k=1,r} (p_j^k) * log_2(p_j^k)$$

The entropy is minimal, equal to 0, if the term t_j appears only in one category. We consider that this term might have a good discriminatory power for the categorization task. Conversely, the entropy is maximal if t_j is not a good feature for representing the documents *i.e.* if t_j appears in all the categories with the same frequency.

We propose a second criteria, denoted CCE (Category Coverage Entropy), combining f_j^k (from CC) and entropy. CCE is defined by:

$$CCE_j^k = (alpha * f_j^k) + (1 - alpha) * (1 - \frac{E_j}{MaxE})$$

where $alpha$ is a parameter and $MaxE$ is the maximal value of E. When the term t_j is characteristic of the category C_k, the value of the criteria is high.

For each category, a subset of the terms of T corresponding to the highest values of the criterion is built. Then, the index is defined as the union of these subsets.

4 Experiments

4.1 Collection INEX XML Mining

The collection is composed of 114 336 XML documents of the Wikipedia XML Corpus. This subset of Wikipedia represents 15 categories, each corresponding to one subject or topic. Each document of the collection belongs to one category. In the XML Mining Track, the training set is composed of 10% of the collection.

4.2 Preprocessing

The first step of the categorization approach that we propose, consists in a preprocessing of the collection. It begins by the construction of the list all the terms (or features) encountered in the documents of the collection. This index of 652 876 terms is build using the LEMUR software[2]. The Porter Algorithm [4] has also been applied in order to reduce different forms of a word to a common form. This operation reduces the index to 560 209 terms. However, it still remains a large number of irrelevant terms that could degrade the categorization, e.g.: numbers (7277, -1224, 0d254c, etc.), terms with less than three characters, terms that appear less than three times, or terms that appear in almost all the documents of the training set corpus. The index obtained at this stage is denoted I. In our experiments, its size is reduced to 161 609 terms on all the documents of the collection and to 77 697 on the training set.

4.3 Features selection

However, as explained in the previous section, the terms of I are not necessarily appropriated for the categorization task inasmuch they are not discriminatory for the categories. This is the reason why our criteria based on entropy and on frequency are used to select more suited features. The terms were sorted according to CC and CCE and only those corresponding to the highest values are retained. In our experiments, the top 100 terms by class and the top 10 000 terms by class were considered for each criteria to build four indexes, denoted respectively CC_{100} and CC_{10000} using CC and CCE_{100} and CCE_{10000} using CCE. Table 1 indicates the size of these indexes.

Index	number of words
I	77697
CC_{100}	1 051
CC_{10000}	75 181
CCE_{100}	909
CCE_{10000}	77 580

Table 1. Indexes sizes

Using one of these indexes, the content of a document is then represented by the $tf.idf$ vector model described in the first section.

The second step is the categorization step itself. Two usual methods of classification are used: Support Vector Machines (SVM) and k-nearest neighbors. Only the most promising results obtained with the SVM were submitted. SVM was introduced by Vapnik for solving two class pattern recognition problems using Structural Risk Minimization principal[7]. In our experiments, the SVM

[2] Lemur is available at the URL http://www.lemurproject.org

algorithm available in the Liblinear library[3] has been used. The results provided by this approach are presented in the next section.

5 Experimental results

This work has been done with a dual purpose: firstly develop a categorization text approach usable as a baseline for further work on XML categorization taking into account the structure, and secondly evaluate performances of this method using our selection features approach.

5.1 Global results

We have submitted 5 experiments using our 5 indexes presented in table 1. The global results of XML Mining 2008 are synthesized in table 2 (participant: LaHC).

Rank	Participant	Run	Recall	Documents
1	LaHC	submission.expe_5.tf_idf_T5_10000.txt	0.7876	102 929
2	LaHC	submission.expe_3.tf_idf_T4_10000.txt	0.7874	102 929
3	LaHC	submission.expe_1.tf_idf_TA.txt	0.7873	102 929
4	Vries	Vries_classification_text_and_links.txt	0.7849	102 929
5	boris	boris_inex.tfidf.sim.037.it3.txt	0.7379	102 929
6	boris	boris_inex.tfidf1.sim.0.38.3.txt	0.7347	102 929
7	boris	boris_inex.tfidf.sim.034.it2.txt	0.7309	102 929
8	LaHC	submission.expe_4.tf_idf_T5_100.txt	0.7230	102 929
9	kaptein	kaptein_2008NBscoresv02.txt	0.6980	102 929
10	kaptein	kaptein_2008run.txt	0.6978	102929
11	romero	romero_naive_bayes_links.txt	0.6813	102 929
12	LaHC	submission.expe_2.tf_idf_T4_100.txt	0.6770	102 929
13	romero	romero_naive_bayes.txt	0.6767	102 929
14	Vries	Vries_classification_links_only.txt	0.6232	102 929
15	Vries	Vries_classification_text_only.txt	0.2444	92 647

Table 2. Summary of all XML Mining results.

5.2 Baseline results

Our baseline corresponds to the first experiment (*expe*_1), which was ranked 3th with a quite good recall: 0.7873.

[3] http://www.csie.ntu.edu.tw/ cjlin/liblinear/ - L2 loss support vector machine primal

5.3 Selection features improves results

When we select 10 000 terms for each class using CCE ($expe_3$) and CC ($expe_5$), we reduce the size of the index to respectively 77 580 and 75 181. This reduction is small compared to the size of the baseline index (77 697). However, it lets us to slightly improve our baseline to 0.7874 with CCE and 0.7876 with CC. These three runs obtained the three best results of the XML Mining challenge.

5.4 Selection features reduces indexes

The last two submitted runs correspond to the selection of the first 100 terms for each class using CCE ($expe_2$) and CC ($expe_4$). As presented in table 1, the size of the index is sharply reduced to 909 terms for CCE and 1 051 for CC. This reduction respectively correspond to 85% and 74% of the size of the baseline index. Even if the obtained results are lower than the results obtained with larger indexes, they are still relatively good. Indeed, the obtained recall is 0.6770 with CCE and 0.7230 with CC.

6 Conclusion

We proposed a categorization text approach for XML documents that let us obtain a good baseline for further work. For now we just used CC and CCE criteria as a threshold to select terms in order to build the index. For future work, we aim at exploiting the computed value of CC and CCE to improve the categorization. Moreover, we could use the structure information of XML documents represented by the links between document to improve even more the results.

References

1. L. Denoyer and P. Gallinari. Report on the xml mining track at inex 2007 categorization and clustering of xml documents. *SIGIR Forum*, 42(1):22–28, 2008.
2. G. Forman, I. Guyon, and A. Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.
3. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning ECML98*, pages 137–142. Springer Verlag, 1998.
4. M. F. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, pages 313–316, 1997.
5. G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.
6. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, 1948.
7. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
8. Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Int. Conference on Machine Learning ICML97*, pages 12–420, 1997.