XML Document Classification using SVM

Samaneh Chagheri, Catherine Roussey, Sylvie Calabretto, Cyril Dumoulin

XML Document Classification using SVM

Samaneh Chagheri*, Catherine Roussey**, Sylvie Calabretto*, Cyril Dumoulin***
*Université de LYON, CNRS, LIRIS UMR 5205-INSA de Lyon
7, avenue Jean Capelle
69621 Villeurbanne Cedex France
samaneh.chagheri,sylvie.calabretto@insa-lyon.fr
** Cemagref, Campus des Cézeaux, Clermont Ferrand
24, avenue des Landais, BP 50085, 63172 Aubière cedex, France
catherine.roussey@liris.cnrs.fr
***CONTINEW
27, rue Lucien Langénieux
42300 Roanne cedex France
cyril.dumoulin@continew.fr

Abstract. This paper describes a representation for XML documents in order to classify them. Document classification is based on document representation techniques. How relevant the representation phase is, the more relevant the classification will be. We propose a representation model that exploits both the structure and the content of document. Our approach is based on vector space model: a document is represented by a vector of weighted features. Each feature is a couple of *(tag, term)*. We have expanded *tf*idf* to calculate feature's weight according to term's structural level in the document. SVM has been used as learning algorithm. Experimentation on Reuters collection shows that our proposition improves classification performance compared to the standard classification model based on term vector.

1 Introduction

The continuous growth in XML documents has caused different efforts in developing classification systems based on document structure. Document representation has to be done before the classification process. Vector space model is one of the most used methods for document representation, in which each document is represented as an N dimension vector of weighted features. N is the number of features in the document collection. The representation models can be divided into three groups: first, the models which do not consider the structure of document focus on representing the document content as a bag of words to classify them. Second are the models which take into account only structure of a document in order to classify them, and third are the models which try to consider both structure and text of XML documents in representation.

Mostly the classification systems use vector space model for document representation, the difference is based on selecting vector features and their weight computation. In (Doucet A., 2002) each vector feature can be a word or a tag of XML document. The *tf*ief (Term Frequency * Inverse Element Frequency)* is used for calculating the weight of words or tags in documents. In (Vercoustre A., 2006) a feature can be a path of the XML tree or a path contacting a word. The term weight is based on *tf*idf.* This allows taking into account either the structure itself or the structure and content of these documents. In (Wisniewski G., 2005) they are interested to only use the document structure for classification without considering the text. They use Bayesien model to generate the possible DTDs of a documents collection. A class represents a DTD. Ghosh (Saptarshi Ghosh, 2008) has proposed a composite kernel for fusion of content and structure information. The paths from root to leaves are used as indexing elements in structure kernel weighted by *tf*idf*. The content and structure similarities are measured individually. A linear combination of these kernels is used finally for a content-structure classification by SVM.

In this article, we propose a method which is an extension of the vector model of Salton (Salton G., 1968) adjusting the calculation of the *tf*idf* by considering the structural element instead of whole document. This representation allows a classification based on content and document structure like the work of Vercoustre, our vector features are different and for calculating their weight we have expanded *tf*idf* in structure level.

The rest of the article is organized as follows. We present our proposition in section 2 on using structure and content on document classification. Experimentation results are written in section 3. Section 4 presents the conclusion and further works.

2 General Overview of our Approach

In this article we have proposed a combination of the content and structure of XML document in the vector model for document representation. In our vector each feature is a couple of *(tag, term)*. Tag corresponds to a structural element in XML document. XML document is represented as a tree. We apply a supervised classification on document collection using Support Vector Machine (SVM). SMV^{light} (Joachims T., 1999) as an implementation of Support Vector Machine has been used for supervised classification.

2.1 Logical structure modeling

We consider the XML document as a tree in which the nodes are tagged by structural labels like title, chapter, etc. The arcs of this tree represent the inclusion relation between nodes, and the leaf nodes contain the document text. The depths of nodes in document tree are important. Thus we consider that two nodes with the same label localized at different depths are two different structural elements. The structural element represents a node type.

We modify document tree organization by aggregating the nodes with the same label and localized at the same depth. For example, all paragraphs in a section are aggregated to a single node “paragraph” which contains all terms of these paragraphs. We take into account only the leaf nodes which contain text for creating document vector. We assume that all documents have a homogeneous structure. There is no node with the same label and different depth. A label becomes an adequate identifier of the structural element.

2.2 Content and structure vector

After aggregating document nodes, we extract the terms in each node. For representing document content a series of linguistic analysis has been done. First the stop words are filtered. Secondly the lemmas of words are extracted using TreeTagger to replace the words. Also, we just keep the words in categories noun and verb which seem more representative. The result of such analysis are called term. Therefore, each feature is made by combining the node label and the term, for example (title: technical) or (p: cleaning). Such feature represents the document structure and content.

For calculating the weight of features we use an extension of *tf*ief* in the document structure level instead of document level and we add the importance of the structural element inside the document. We assume that how deeper a node is in the document tree, less it will be important. So, the weight of feature will be as shown below:

$$W_{i,e,d} = TF_{i,e,d} * IDEF_{d,e} * IED_e$$

i, *e*, and *d* represent respectively a term, a structural element type like a paragraph and a document in the collection.

The Term Frequency $TF_{i,e,d}$ is the number of occurrences of the term *i* in structural element type *e* inside the document *d*, normalized by the number of all terms in this node. The Inverse Document Element Frequency $IDEF_{d,e}$ is calculated by $log\frac{D_e}{d_{e,i}}$ with D_e as the number of documents in the collection having a node of type *e*, and $d_{e,i}$ as the number of documents having node *e* containing the term *i*.

The Inverse Element Depth IED_e represents the importance of a structural element *e* in the collection. It is calculated as $log\frac{L_d+1}{l_{d,e}}$ with L_d as the average depth of documents tree in the collection, and $l_{d,e}$ as the average depth of the node of type *e* in the collection.

After constructing documents vector, we apply SVM as classification algorithms for learning and classifying our collection.

3 Experimentation

Experiments were performed on the Reuters Corpus Volume 1 (RCV1) which includes over 800,000 English language news stories. The stories are formatted using a consistent XML schema. In this collection we use only header and paragraph of stories which are all in the same depth in all documents. Each story is annotated for topic, region and industry sector and they are multiclass. We have ignored unclassified documents and for mono classification we have selected the first class of topic category of each story. We have made a learning collection of 400 documents with 200 positive and 200 negative examples for a topic class called “GCAT” and a test collection with 400 documents. We made two experimentations using SVM. The first one called “content and structure” is an implementation of our proposition. The second one called “content only” uses a vector of simple terms weighted by standard *tf*idf*. The results of the classifications are shown in table 1. The results demonstrate that including structure improves the classification accuracy.

Vector features	Precision	Recall	F-measure
Content & structure	0.92	0.96	0.94
Content only	0.93	0.83	0.88

TAB1. Precision/Recall results on Reuter corpus

4 Conclusion and Perspectives

In this article we have proposed a model for XML document representation in order to classify them. We proposed a combination of structure and content in document representation by the vector model. Our features are couples of *(tag: "term"),* all weighted by an extension of *tf*idf* taking into account the terms position in the documents tree.

We have done our experimentation on Reuters XML news collection which is a particular collection with homogenous documents. In future works we are going to improve our proposition in order to do the test on a heterogeneous corpus like INEX evaluation campaign collection which contains a huge number of documents with heterogeneous structures.

References

Doucet A., A.-M. H. (2002). Naive Clustering of a Large XML Document Collection. *INEX Workshop 2002* , 81-87.

Joachims T. (1999). Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press* .

Salton G. (1968). Search and retrieval experiments in real-time information retrieval. (C. university, Ed.) 1082-1093.

Saptarshi Ghosh, P. M. (2008). Combining Content and Structure Similarity for XML Document. *ICPR* , 1-4.

Vercoustre A., F. M. (2006). Classification de documents XML à partir d'une représentation linéaire des arbres de ces documents. *INRIA 2006* .

Wisniewski G., D. L. (2005). Classification automatique de documents structurés. Application au corpus d'arbres étiquetés de type XML. *CORIA 2005 Grenoble* .

Résumé

Le présent document décrit une représentation pour les documents XML basée sur le modèle vectoriel. Les composants de vecteur sont les couples pondérés de *(balise, terme).* Nous avons élargi *tf * idf* pour calculer le poids selon le niveau structurel de terme dans le document. SVM a été utilisé comme l'algorithme d'apprentissage. Expérimentation sur le corpus Reuters RCV1 montre que l'ajout de la structure au vecteur de document améliore la performance de classification par rapport au vecteur de terme.