UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

A BASELINE APPROACH FOR THE AUTOMATIC HIERARCHICAL ORGANIZATION OF LEARNING RESOURCES

Paramjeet S.Saini, Diego Sona and Marco Ronchetti

January 2004

Technical Report # DIT-04-006

A Baseline Approach for the Automatic Hierarchical Organization of Learning Resources

Paramjeet S.Saini
University of Trento
Dept. of Computer Science and Telecommunications
Via Sommarive 14, Povo (TN), Italy
pssaini@dit.unitn.it

Diego Sona
ITC-irst
Automated Reasoning Systems Division
Via Sommarive 18, 38050 POVO (TN), Italy
sona@itc.it

Marco Ronchetti
University of Trento
Dept. of Computer Science and Telecommunications
Via Sommarive 14, Povo (TN), Italy
marco.ronchetti@unitn.it

Abstract: Over the past decades we have seen the exponential growth of learning resources on the web, with thousands of documents easily available. As the availability of learning resources increases, the difficulty to find required and relevant learning resources becomes more and more apparent. We believe that hierarchical organization of web learning resources, according to some predefined concept hierarchy, could solve the above mentioned problem up to some extent. Manual classification of learning resources is a tedious task, and mechanism to automate classification process is in high demand. This paper discuses the approach we have designed for the automatic classification of computer science learning resources.

1 Introduction

In the past years, rapid evolvement of the Web and its application like E-Learning had tremendous influence on the universities and higher education institutes. In the recent years we have seen a rapid growth in the number of learning resources available on the Web. Manufacturing of learning material is both a time consuming and expensive task: it would therefore important to be able to reuse existing learning resources. Moreover, students would benefit if they were able to retrieve during learning sessions learning materials that are suited for their needs even when provided by additional, external sources. Due to massive information overload on the Web, the main problem here is to find and manage as much as possible the relevant learning resources.

We believe that hierarchical organization of learning resources could face the above-mentioned problem. Hierarchical categorization according to some predefined categories or classes proved to be very useful in the E-Learning domain, where the topic's to be learned are organized in the form of concept hierarchy. However, manual filtering and classification are time consuming and expensive tasks. As a result, we need a mechanism to automate the classification process. This paper discuses a classification model that we devised to automate the classification process.

The paper is structured as follows: section 2 introduces the preparation of hierarchy of concepts used to test the model, section 3 gives a brief introduction to the hierarchical classification task and its advantages in E-Learning domain, and in section 4 we describe the approach we followed for classification and the overall architecture of classification model. Finally, in section 5 we conclude giving also an idea of some directions for future work.

2 Preparing the Concept Hierarchy

Any learning resource can be defined in terms of related concepts. To deliver clear concepts about any of the subject's area it is necessary to find precise relationships between documents belonging to different concepts. This goal can be attained if the learning resources to be delivered are arranged according to a predefined concept hierarchy. A concept hierarchy, actually, is an efficient way to organize, index, and explore the available knowledge.

In practical settings, performance of classification models depends on how much descriptive is the provided concept hierarchy. To create a concept hierarchy spanning all the aspects of computer science we need for an ontology describing the computer science domain in an exhaustive way. To achieve this task we derived an ontology (Ronchetti, 2003) from the ACM Computing Curricula 2001 for Computer Science (for details refer http://www.computer.org/education/cc2001). The extracted ontology consists of 14 areas, 132 units and 950 topics (see table 1) below.

Area	Units	Topics
Discrete Structures	6	45
Programming Fundamentals	5	32
Algorithms and Complexity	11	71
Architecture and Organization	9	55
Operating Systems	12	71
Net-Centric Computing	9	79
programming languages	11	75
Human-Computer Interaction	8	47
Graphics and Visual Computing	11	84
Intelligent Systems	10	106
Information Management	14	93
Social and Professional Issues	10	46
Software Engineering	12	85
Computational Science	4	61

Table 1: Subject Area's covered in the Ontology

We generated the concept hierarchy (see figure 1) according to the areas covered in the above-mentioned ontology (see table 1). Using these areas we were able to generate a hierarchy with sufficient descriptive knowledge for all nodes. Manual extraction of concept hierarchy from above mentioned ontology is quite time consuming and tedious task. Therefore, we designed a tool to automate the process of concept hierarchy generation. At present, the tool is embedded within the classification model and it generates the concept hierarchy on the fly. Clearly, we could save processing time by creating the hierarchy only once, but current architecture allows us to add new concepts at any time.

In the currently generated concept hierarchy there are 14 nodes at first level, 132 sub-nodes and more then 1500 class representatives (features or keywords). A partial view of the derived concept hierarchy is shown in figure 1. For each node in the concept hierarchy there is sufficient knowledge (class representatives) that allows the classifier to determine the relevant locations of a given document in the hierarchy. In particular, the task here is to organize the learning resources within the hierarchy of concepts, allocating the document in the leaf nodes. Hence the classification task needs to decide the most relevant leaf for any input document, exploiting the hierarchical information, i.e. area, unit, and topic information.

3 Hierarchical Classification

Hierarchical classification of documents is a task that received growing interest in information retrieval (IR) and machine learning (ML) communities, due to the widespread proliferation of topic hierarchies for

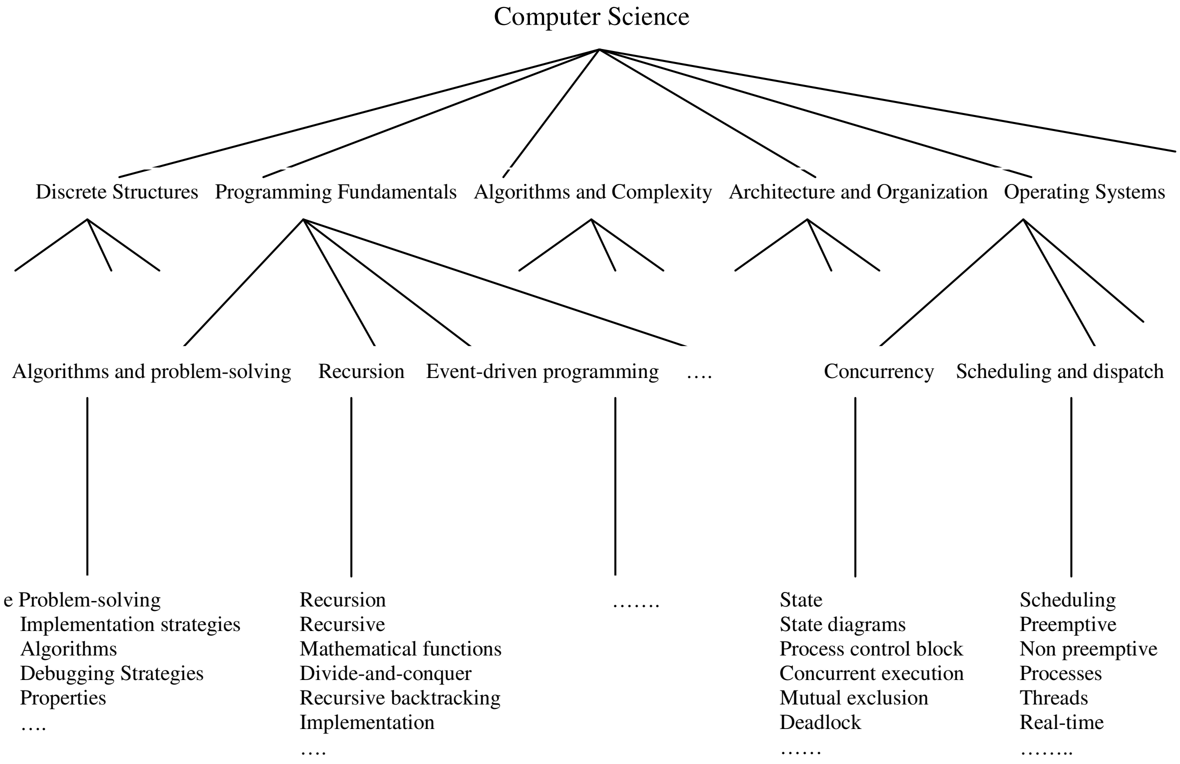

Figure 1: Partial View of the Concept Hierarchy

text documents. Specifically, classification (Chun-hung, 2001) is a process driven by a function that associates an unlabeled object (in this case an unclassified learning resource) to one or some classes from a given set of possible classes. Therefore, the classification process deals with assignment of learning resources to a set of predefined categories, where a learning resource may have some relations with other categories in the concept hierarchy. For example a document related to "Propositional logic" could be assigned to the class associated to the concept "Basic logic", which in turns belongs to the class "Discrete Structures", which in turns belong to the class "Computer Science". Whenever all the information coming from the different level of hierarchy (in our task: area, unit, and topic) is used to classify a document, we can talk of hierarchical classification. The exploitation of the hierarchical knowledge has proven to be very useful in classifying resources; therefore, the extension of the approach to the organization of E-learning documents within a hierarchy of concepts is natural and straightforward.

In the area of hierarchical document classification lots of works have been done within the Information Retrieval and Machine Learning communities, some examples can be found in (Axin, 2001), (Ceci, 2003), (Cheng, 2001), (Koller, 1997), (Dumais, 2000), (Ruiz, 2002), (Wang, 1999). All the proposed models are based on supervised learning strategy. In supervised learning setting, classifiers learn from a set of training data, which is composed of labeled examples. New documents are then classified on the basis of the experience acquired by the classifier during the training phase.

Our task, on the other side, is slightly different from a classical supervised task. Actually, documents need to be classified without the knowledge of already labeled examples (Adami, 2003). Basically, documents need to be classified just using the knowledge embedded in the hierarchy, i.e. the set of keywords associated to the nodes in the hierarchy (see the example of figure 1). For this reason we devised a basic solution to classify documents using such keywords. Next section discusses the approach we followed to develop concept hierarchy. For experiments purpose, we began our task with the classification of computer science learning resources.

4 Our Approach for Classification

We observed that, to develop a hierarchical classification model we need two forms of knowledge. Firstly, we need a concept hierarchy according to which learning resources could be classified and secondly we need document vocabulary or keywords that can give significant description of the learning resources. After the determination of the vocabulary we can then classify documents according to a given concept hierarchy. Specifically, the classifier compares the documents with the class descriptors and it labels the documents with the class where a suitable match is found.

It's well known that the string comparison is quite a time consuming task that can affect the overall performance of model. To overcome this problem we represented the concept hierarchy in terms of set of binary vectors. For each node in the concept hierarchy a binary vector describing the related concept is determined filling the vectors according to the presence or absence of the keywords describing the concept. Each position in the vectors represents a given keyword in the vocabulary.

To classify learning resources, documents are also represented in terms of binary vector. The classification task is then based on a comparison between document representations and concept descriptors, and documents are labeled with the node where most appropriate match is found.

4.1 Extracting Document Vocabulary

Our preliminary approach for the determination of the documents' vocabulary was based on existing machine learning tools and techniques (Ronchetti, 2004). Specifically, we determined the vocabulary of the documents' dataset extracting the most representative keywords with a machine learning system known as Kea (Witten, 1999). Kea is an algorithm that extracts key-phrases from a document using a naive Bayes approach. We observed that the performance of the classification model was heavily biased by the output of Kea. The main problem was that sometimes Kea's output is a set of terms that can not be used by the classifier (because they are not at all present in the concept hierarchy). As a result documents are discarded because the classification is impossible.

To overcome the above-mentioned problem we modified our previous model by replacing Kea with our own document vocabulary extractor. Specifically, the vocabulary extractor only uses those terms that are found in the main ontology.

4.2 Overall Architecture

In Figure 2 are shown the main ingredients of the classification architecture, which are:

Web Crawler: that retrieves the learning resources from directed learning domain, initially we supply some addresses of the learning domains from where the system can find some learning material.

Html2txt Converter: that removes html tags and performs some basic cleaning. The output is plain text.

Ontology: to achieve good performance during classification, concept hierarchy should be descriptive enough of the domain. We generated the concept hierarchy according to the areas described in above-mentioned ontology so that the hierarchy is sufficiently descriptive of the domain. Our classification models rely heavily on the descriptive power of the ontology (Ronchetti, 2003).

Cleaner: that performs two main functions: it removes all stop words and then it checks for repeated terms. The output from is a list of keywords without repetitions and stop words.

Document Vocabulary: is the set of keywords extracted by the DV extractor module from the supplied document. Only those terms that are present in the standard vocabulary are extracted.

Standard Vocabulary: is the bunch of computer science domain terms generated by performing cleaning of the ontology terms. At present there are 1500 terms. Standard vocabulary is used to determine the document and the concept encodings.

Concept Hierarchy Generator: the input to the concept hierarchy generator is the cleaned ontology (Ronchetti, 2003), while the output is the concept hierarchy (in figure 1 is shown a part of the hierarchy. Each node in the hierarchy consists of a node name and a list of keywords (class descriptor) that identify the set of features used to determine the fitting measure.

BV Generator: generates the binary vector corresponding to the input file that consists of the domain terms. To generate binary vectors of input file it takes one term at a time and checks its presence in standard vocabulary.

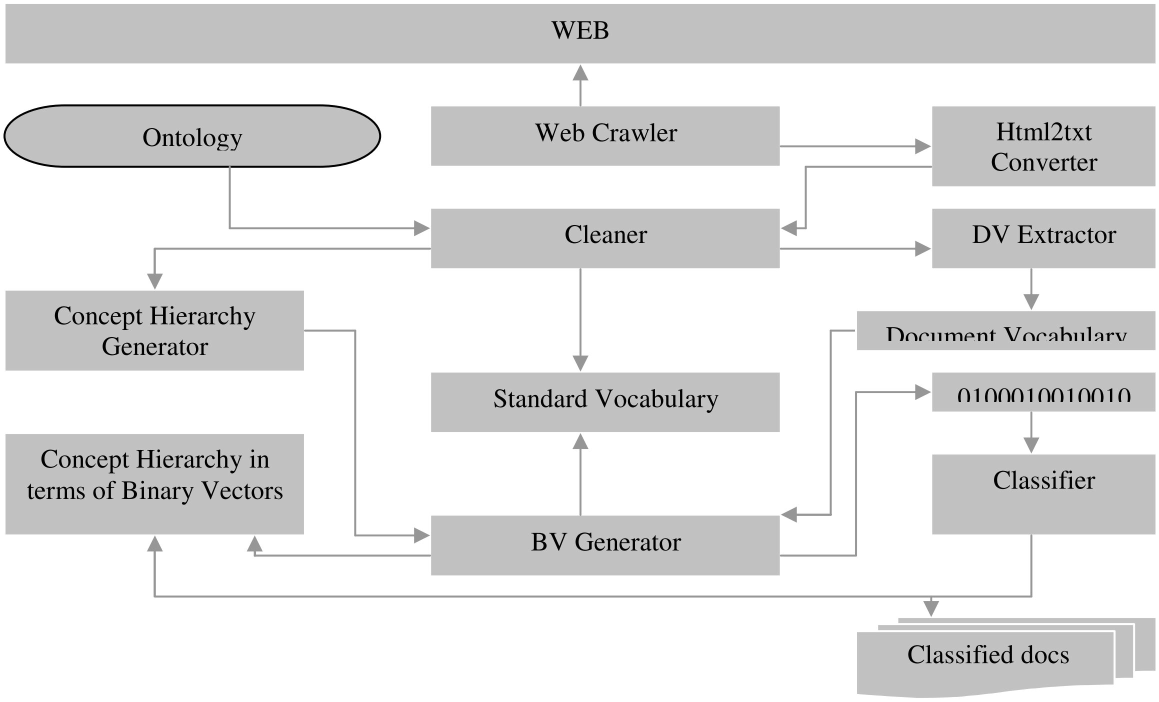

Figure 2: Architecture for Classification.

Classifier: To classify document according to the concept hierarchy, the classifier takes the document encodings (represented in terms of binary vector) and find the most appropriate class label. The simple classification mechanism is based on a greedy approach. In the first step the module compares the document with the class descriptors, and different weights are assigned to each of the 14 nodes. In second step, the node with the highest weight is selected and all sub-nodes are compared with the document. The sub-node with the highest resulting weight is designated as the document label. Following this top down approach for classification could also result in exploring other topics in the concept hierarchy that are related to classified document.

5 Conclusions and Future Work

In this paper we presented a working model for the automatization of the classification process. The model organizes learning resources according to the concept hierarchy that we have extracted from an ontology. The performance of classification model relies heavily on the descriptive power of ontology. Up to some extent, the architecture is capable of addressing search and retrieval problem of e-learning resources.

Presently our system is suffering two main problems. Firstly, the standard vocabulary extracted from the ontology consists of only 1500 domain terms. The number seems to small for a sufficient covering of all the topics of the computer science domain. Moreover, document encodings do not use some important terms which are not included in the standard vocabulary. Because of the lack of strong descriptive knowledge, sometimes documents are wrongly classified. Secondly, there are 1500 features describing the domain, but the number of features for each node is quite a few. After reaching the final node in the classification process if classifier can’t find the proper document vocabulary match then our model simply discards that document. The possible solution to these problems could be to create a mechanism able to extract important domain specific terms form already classified documents. These new terms can be added to standards vocabulary and can provide a more detailed description for each node in the concept hierarchy and for the domain as a whole.

References

Aixin Sun, Ee-Peng Lim (2001), Hierarchical Text Classification and Evaluation, In ICDM 2001, IEEE Int. Conf. on Data Mining.

Ceci M and D.Malerba. (2003). Web-pages classification into a hierarchy of categories. In Proc. of the 25th European Conf. on Information Retrieval (ECIR'03).

C.H.Cheng, J. Tang, A. Wai-chee, and I. King (2001), Hierarchical Classification of Documents with Error Control, In PAKDD 2001, 5th Pacific-Asia Conf. on Knowledge Discovery and Data Mining.

Chun-hung C, Jian T, Ada W, Irwin K, (2001).,"Hierarchical Classification of Documents with Error Control ".Proceedings of PAKDD-01, 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining.

D. Koller, M. Sahami (1997), Hierarchically Classifying Documents Using Very Few Words, In ICML 1997, 14th Int. Conf. on Machine Learning.

M.E. Ruiz and P. Srinivasan (2002), Hierarchical Text Categorization Using Neural Networks, In Information Retrieval Vol. 5(1).

Nigam, Kamal; Maccallum, Andrew Kachites.(1999).; Thrun, Sebastian and Mitchell, Tom. Text Classification from Labeled and Unlabeled Documents using EM. The Machine Learning Journal 1999. Draft.

Ronchetti, M. and Saini, P.S. (2003). "Ontology-based metadata for E-Learning in the Computer Science domain", IADIS e-Society 2003 Conference.

Ronchetti, M. and Saini, P.S. (2004). "Machine Learning: an Approach for Classifying Learning Resources". To appear in 2004 IRMA INTERNATIONAL CONFERENCE, May 23-26, 2004.

S. Dumais and H. Chen (200), Hierarchical Classification of Web Document, In Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR'00)".

Witten I.H., Paynter G.W., Frank E., Gutwin C. and Nevill-Manning C.G. (1999) "KEA: Practical automatic key-phrase extraction." Proc. DL '99, pp. 254-256.