Thesaurus topic assignment using hierarchical text categorization

Franciso J. Ribadas
Dept. de Informatica
Universidade de Vigo
Campus de As Lagoas, s/n, 32004
Ourense, Spain
ribadas@uvigo.es

Erica Lloves
Telemaco, I. D. S., S.L.
Parque Tecnologico de Galicia
San Cibrao das Vinas
Ourense, Spain
ericalloves@gmail.com

Victor M. Darriba
Dept. de Informatica
Universidade de Vigo
Campus de As Lagoas, s/n, 32004
Ourense, Spain
darriba@uvigo.es

ABSTRACT

In this paper we present a method for assigning topics from a hierarchical thesaurus to documents written in natural languages. The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized. In our case the classification problem is reduced to a sequence of partial classifications, guided by the structure of the topic tree, using a specific set of features at each node in the hierarchy.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*linguistic processing, thesauruses*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

General Terms

Documentation, Design

Keywords

Natural language processing, text categorization, thesaurus

1. INTRODUCTION

In many document processing tasks a correct identification of relevant topics offers a helpful starting point to develop advanced applications to deal with browsing and searching in large collections of documents. In this context, one of the most valuable tools are specialized thesauri. These kinds of structure organize a set of concepts relevant to a given domain in a hierarchical structure, making it possible to employ a sort of controlled vocabulary to simplify document processing.

SIGIR'07 iNEWS07 workshop,
July 27, 2007, Amsterdam, The Netherlands.

The classical approach [2] relies on human processing to perform thesaurus term selection after reading each document in the collection. This approach requires the availability of trained experts and suffers from a lack of scalability, since this kind of work is very time consuming and difficult to apply on large collections of documents. We propose to partially replace this kind of human made task with an automatic tool able to identify, for each input document, a list of potential descriptors taken from the domain thesaurus.

In this paper we describe our preliminary work on the automatic assignment of relevant topics, taken from a structured thesaurus, to documents written in natural languages. In our case we are interested in the domain of legislative texts in Spanish. We have an available thesaurus, manually built, with more than 1800 concepts, arranged in a tree structure. We also have a collection of legislative documents, whose main topics have been identified by humans according to the entries in that thesaurus.

The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized. Many previous proposals [11] have dealt with text categorization, but the case of hierarchical classes is usually omitted [8] or the generalization to multiple label classification is not directly supported [3, 4].

Our aim is to build a system able to assign descriptive topics to input documents. The set of possible topics is taken from the thesaurus entries and for each document many descriptors may be selected with no special restrictions about the relationships among them. So, in the set of assigned descriptors we could find pairs of sibling entries or any combination of ancestors and descendants.

We also want our system to model in some way the processing made by humans when they perform this kind of task. In our domain, legal texts in Spanish, a very restricted kind of document structures is commonly employed. Document contents can be segmented into consistent text regions and expert users pay special attention to those specific portions which usually carry the most relevant content. Examples of these are the document introduction, the description of the document aims, the destination section or text portions

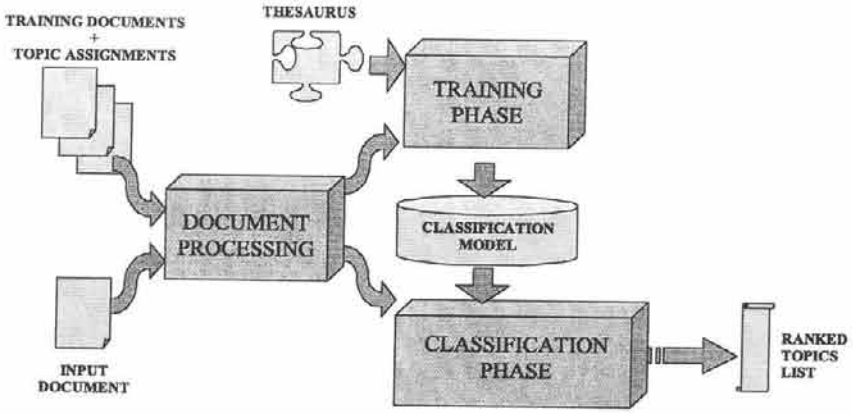

Figure 1: Classification framework.

dealing with legal motivations and background. Also, human experts tend to use the thesaurus structure as a guide to select descriptive topics from it. So, we maintain this two intuitions in our approach that filters entries from the topics hierarchy in a top-down fashion.

The article is outlined as follows. Section 2 introduces our classification framework. Next, Section 3 describes the document representation and processing. In Section 4 the most relevant details about the training and classification strategies are described. Section 5 shows the results obtained in our preliminary experiments. Finally, Section 6 presents our conclusions and future work.

2. TOPIC ASSIGNMENT AS A CLASSIFICATION TASK

Some questions need to be taken into account before starting to describe our proposal. First of all, we are working on a big domain from a text processing point of view. On the one hand, we have a very large collection of legislative documents. These documents tend to be quite long, with sizes ranging from hundreds to thousands of words. On the other hand, we also have a big set of potential classes arranged in a tree. In this context there are two main aspects to have in mind. First of all, we must ensure a practical computational cost, both in the training phase and specially in the topic assignment phase. We also must offer a robust classification framework, able to return a consistent list of topics for a great variety of input documents, without contradictions. With regard to the available resources, we have a thesaurus built by hand for the domain of legislative text and a set of historic documents with their corresponding descriptors assigned by human experts, which will be employed in the training phase.

With these premises in mind, a first approach could be to take the available documents and train a big classifier using all of the topics in the thesaurus as output classes. This approach is almost impractical from a computational cost point of view, but also it has many important problems with output quality and a lack of robustness and consistency. Training such a classifier involves estimating a large number of parameters with too many irrelevant features that will disturb the classification decision.

The strategy we have chosen is inspired by Koller and Sahami's work [6] and takes advantage of the class hierarchy to simplify the classification task in two aspects. Firstly, the global classification problem is reduced to a sequence of partial classifications, guided by the structure of our topic tree. Secondly, the computational cost for each classification step is reduced and the resulting quality is improved by means of the use of a specific set of features, exclusive to each node in the hierarchy. In this manner, the classification decision is distributed over a set of partial classifiers across the topics tree. In this model each internal node will be responsible for a local classification decision, where only a small set of features from the document will be taken into account to select the most promising descendants, where this processing will be repeated.

The main difference from Koller and Sahami's proposal is the final output of our classifier. Our aim is to get a set of relevant topics taken from the thesaurus, ordered according to their relevance, instead of a single class. We replace the greedy approach used in the original work, where only the best successor for each node was considered at each step. In our proposal, we proceed level by level, and all of the paths starting at successors with higher evaluation values are taken into account, and they are followed until no further expansion is admissible. The final set of topics is composed of those class nodes, ranking them according to the strength of the classification steps that lead to them.

In Fig. 1 we show the main phases in our approach, where three main components are outlined. Document processing, which is applied both in training and classification, has the responsibility of cleaning the documents to reduce the number of features employed to represent them. In the training phase, the training set of documents are taken and the topic hierarchy is traversed top-down, performing at each node local feature selection and training the corresponding partial classifier. Finally, in the classification phase, for each input document the topic tree is traversed top-down using the trained classifiers to decide whether the corresponding topic is suitable to be taken as a final descriptor and to make routing decisions to select one or more branches to continue searching. At the end, the list of potential descriptors for that document will be ranked and returned to the user.

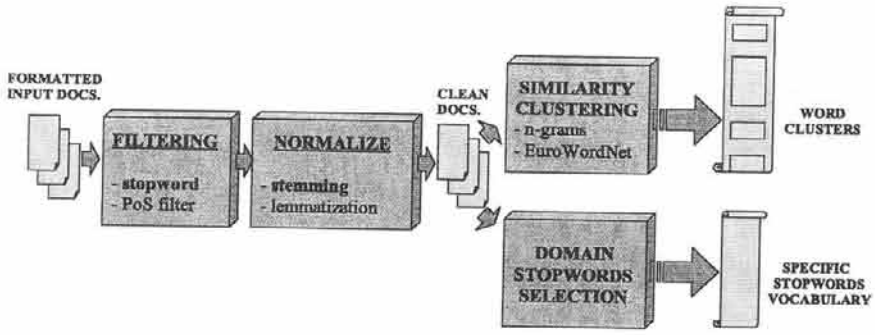

Figure 2: Collection preprocessing.

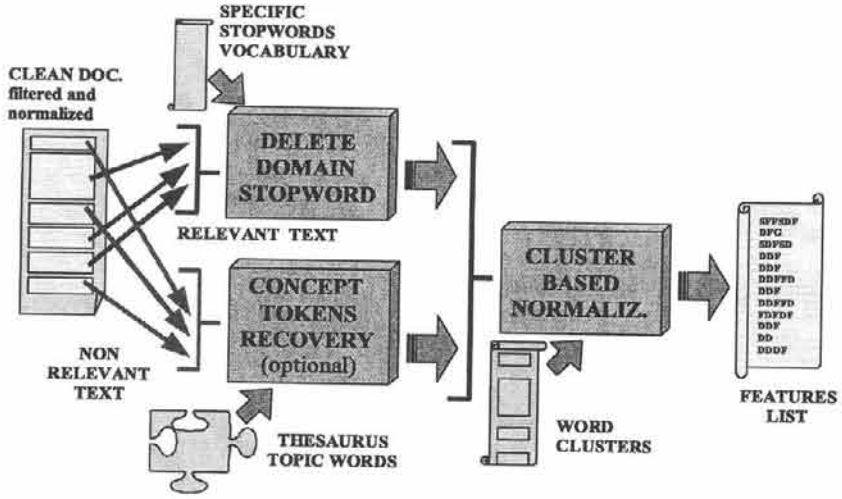

Figure 3: Document processing and representation.

3. DOCUMENT PROCESSING

Since this is a first approach to this kind of problem, we have tried to avoid using complex linguistic resources, like taggers, lemmatizers or shallow parsing [12]. Original documents were in HTML and PDF format and the first step was to extract plain text from them. Those text files were previously preprocessed to segment their text into regions and to identify which of those regions are relevant and could be suitable to extract potential descriptors from them.

A first processing, shown in Fig. 2, is performed on the whole collection. To omit non-relevant words we use a generic stopword list for Spanish. Remaining words are normalized by means of stemming rules to overcome lexical variation problems. Once all of the the documents in the collection have been cleaned, two structures are built. A specific stopword list, containing a vocabulary of commonly used words in the considered domain, makes it possible to get rid of words frequently employed in legislative texts. A dictionary of similar words, that allows us to identify groups of related words, is also built. We have employed a method to detect similar tokens at orthographic level by means of a hierarchical clustering algorithm which uses a n-gram based distance [7] between word characters.

Both in training and classification, the list of features to be employed is extracted from cleaned documents in the way shown in Fig. 3. From the relevant regions of the input document, domain specific stopwords are deleted. Optionally, some words that appear as labels in the thesaurus topics can be recovered to be taken into account as features. These features have been demonstrated to be useful when short documents are processed. The surviving features will undergo a sort of semantical normalization using the similar word clusters built from the whole training collection. After that we obtain the list of features that will describe the input document in the training and classification phases.

4. TRAINING AND CLASSIFICATION

In this section we show the main components that comprise our approach. Once the set of training documents have been processed they are employed to train our hierarchical categorization model. This model contains for each thesaurus node the set of features with higher discrimination power at that level and a trained partial classifier to make the routing decisions. The idea behind this strategy is that using this set of classifiers' decisions will be more precise and the overall classification quality will be improved.

4.1 Training the hierarchy

In the training phase we take the whole training collection with the set of topics associated to each document and we

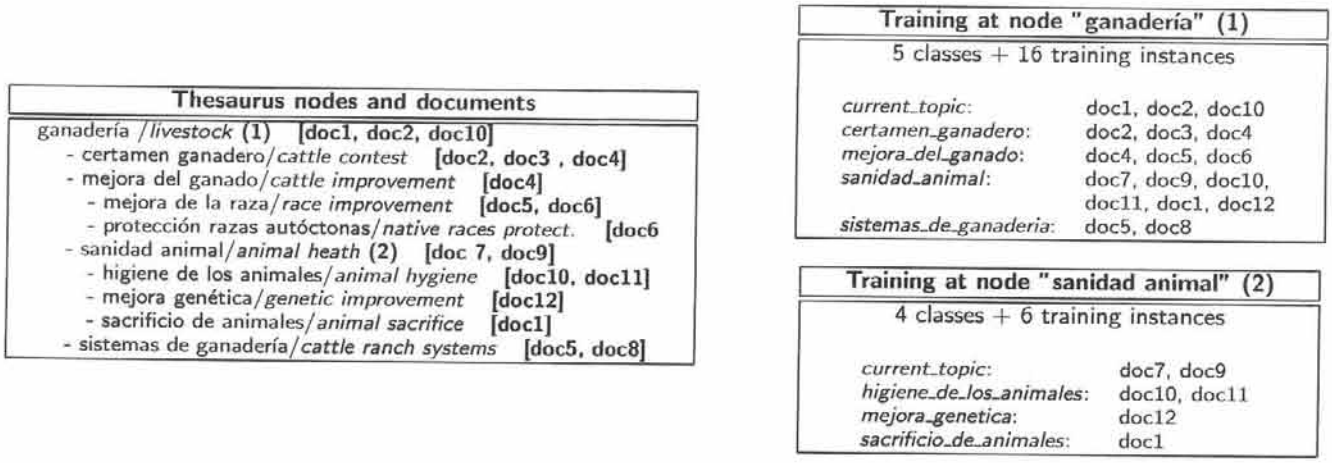

Figure 4: An example of training at two nodes.

traverse the topic tree, performing two tasks at every thesaurus entry. For each node, the subset of documents with at least one descriptor being a descendant of the current concept is selected. With these documents we apply very simple feature selection techniques to find a set of features with the highest discrimination power among the different branches starting at this node.

The actual feature selection method employed in our system is controlled by two thresholds, **Th1** and **Th2**, and works as follows:

1. The set of classes for the current node, $\mathcal{C}$, is built.
   - One class will correspond to the topic at the current node, which will be associated with documents having that topic as a descriptor.
   - For every direct descendant of the current topic another class is defined, which will be associated with documents having at least one of the descriptors belonging to the branch starting at that descendant, as shown in Fig. 4.
2. For each class $C_i \in \mathcal{C}$:
   - Every word w_{ij} in a document j associated with C_i is inspected.
   - Word w_{ij} will survive feature selection if:
     (a) w_{ij} is present in AT LEAST **Th1** % of documents being associated with class C_i
     (b) w_{ij} is present in NO MORE than **Th2** % of documents not being associated with class C_i

Once the external feature selection is performed, a specialized classifier is trained to select the most promising branches at the current level. For each document a feature vector is built. Only the relevant stems selected for the current concept are employed, using their *tf-idf* [10] as feature values. The class for this training vector will be the current topic, if it is actually associated with the document, or the label of one of its sons, if some topic associated with the document falls into that branch. Fig. 4 illustrates this idea. We have employed the WEKA machine learning engine [13] to train the specialized classifiers for each topic in our hierarchical thesaurus. We have tested several classification algorithms to be employed inside this hierarchical categorization scheme, as it can be seen in the experimental results section.

4.2 Hierarchical classification

Once all of the partial classifiers have been trained, the assignment of topics to new documents means traversing the thesaurus tree, as shown in Fig. 5. Starting at the thesaurus root, the feature vector for the document is built using the selected features for each node, and the most promising branches according to the partial classifier results are followed.

The original proposal by Koller and Sahami defines a single class output. They perform a greedy search selecting at each node only one class and stopping when a leaf is reached. Since we are interested in multilabel classification, we have added two new components in our classification strategy. In this way, node classifiers have two missions. The first one is to detect if a topic is suitable to be considered as a final descriptor, and the second one is to make a routing decision to determine the next steps in the search.

The routing decisions taken at each node are controlled by simple thresholds that take into account both the number of alternatives at each node and the strength of the potential classes returned by the classifier. If the class for the current topic has an evaluation value higher than this threshold it is considered to be suitable as a topic for describing this document. When a leaf is reached or no successor classes have sufficient strength, the deeping is stopped. The final list of potential topics is ranked according to the set of values obtained in the sequence of partial classifications that lead to them. Different formulae, average, maximum or product, can be used to combine the strength values obtained in the path of classifications from the root to that descriptor.

5. EXPERIMENTAL RESULTS

To illustrate our proposal we will review some preliminary experiments we have performed to test our method. In these

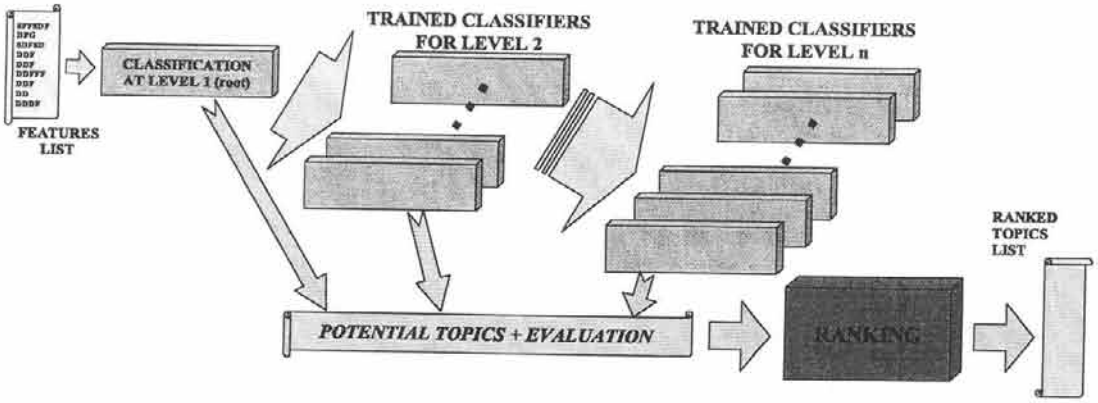

Figure 5: Hierarchical classification.

experiments we have employed a portion of the legal corpus donated by Telemaco, S.L., with 2333 legislative documents with their corresponding set of descriptors assigned by human experts. These descriptors where taken from a set of 1873 thesaurus topics about the fields of agriculture, livestock and fishing. This corpus was randomly split to build a training data set with 2124 documents and a test dataset with 209 documents. To evaluate the experimental results we have employed two well known measures in the Information Retrieval field, precision and recall, using a modified version of the standard `trec_eval` [1] tool to compute them.

In the experiments reported in this paper we have evaluated two aspects in our proposal. Firstly we have tested the influence on the final results of different approaches to generating the input text. Secondly we have evaluated the suitability of several text classification algorithms. Fig. 6 shows the results obtained using different text sources from the original documents to extract features from them. We have taken words only from the document title (experiment [T]), words from the title and the relevant regions (experiment [T+RR]) and we included selected words taken from non-relevant regions, giving them different weights (experiments [T+RR+SW] and [T+RR+2SW], where selected words count twice). As can be seen in Fig. 6 the best results were obtained using words from the title, words from relevant regions and selected words from non-relevant ones.

In Fig. 7 we show the average precision and recall values obtained in a set of experiments to test the use of different machine learning algorithms to perform the partial classifications across the thesaurus tree. We have tested a Naive-Bayes implementation [5], a k-Nearest Neighbors(k-NN) [1] learning method, with different values for k, and a Support Vector Machine model using Sequential Minimal Optimization(SMO) [9], all of them are included in the WEKA machine learning engine [13]. As can be seen, the best results, both in precision and recall, where obtained with the k-NN method, with a better balance between absolute recall and precision when seven neighbors were employed. In a deeper review of the descriptors obtained in that run, our approach gave better results when dealing with the most general topics, but it was unable to get a human level performace with the most specific descriptors.

[1] `http://trec.nist.gov/trec_eval`

6. CONCLUSIONS AND FUTURE WORK

In this article we have proposed the use of a hierarchical multilabel classification approach which allows us to face the thesaurus topic assignment problem. We have followed a very flexible method, easy to be adapted to deal with different practical domains and allowing the use of several classification and text processing algorithms. The developed system offers quite good performance on average documents, even being able to avoid some human inconsistencies. When complex or very specific documents are processed, our tools are unable to work at human expert level, opening a field for further improvements.

With respect to future work, several aspects should be studied in our classification approach. Firstly, we intend to extend our experiments to other domains and languages, in order to test its generality. Secondly, we aim to improve the system by integrating more powerful natural language processing tools.

Acknowledgements

The research reported in this paper has been partially supported by Telémaco, Información, Documentación y Sistemas, S.L. (`http://www.telemaco.com`), Xunta de Galicia (PGIDIT05SIN044E, PGIDIT05PXIC30501PN), Ministry of Education and Science (TIN2004-07246-C03-01) and University of Vigo

7. REFERENCES

[1] Aha, D., and D. Kibler. Instance-based learning algorithms. *Machine Learning*, vol.6, pp. 37-66, 1991.

[2] J.H. Choi, J.J. Park, J.D. Yang, and D.K. Lee. An object-based approach to managing domain specific thesauri: semiautomatic thesaurus construction and query-based browsing. *Technical Report TR 98/11*, Dept. of Computer Science, Chonbuk National University, 1998.

[3] W. Chuang, A. Tiyyagura, J. Yang, and G. Giuffrida. A fast algorithm for hierarchical text classification. In *Proc. of the 2nd Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK'00)*, pp 409-418, London, UK, 2000.