Automatic Indexing from a Thesaurus Using Bayesian Networks: Application to the Classification of Parliamentary Initiatives

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Alfonso E. Romero

Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicaciones, Universidad de Granada
18071 – Granada, Spain
{lci,jmfluna,jhg,aeromero}@decsai.ugr.es

Abstract. We propose a method which, given a document to be classified, automatically generates an ordered set of appropriate descriptors extracted from a thesaurus. The method creates a Bayesian network to model the thesaurus and uses probabilistic inference to select the set of descriptors having high posterior probability of being relevant given the available evidence (the document to be classified). We apply the method to the classification of parliamentary initiatives in the regional Parliament of Andalucía at Spain from the Eurovoc thesaurus.

1 Introduction

To improve organizational aspects and facilitate fast access to relevant information relative to a particular subject, document collections from many organizations are classified according to their content using a set of descriptors extracted from some kind of controlled vocabulary or thesaurus. For example, most of the parliaments in Europe use a thesaurus called Eurovoc to classify parliamentary initiatives, the Food and Agricultural Organization (FAO) employs Agrovoc to categorize its documents, and the National Library of Medicine (NLM) uses MeSH to index articles from biomedical journals. The process of assigning descriptors in the thesaurus to the documents is almost always carried out manually by a team of expert documentalists. The objective of this work is the development of a computerized tool to assist the human experts in this process.

So, the scope of the paper is automatic subject indexing from a controlled vocabulary [6,10] and hierarchical text classification [11,14]. However, given the critical nature of this task in many contexts, it is not realistic to try to design a completely automatic classification process, and final human supervision will always be required.

An important characteristic of the model that we are going to propose is that no training is required. We shall exploit only the hierarchical and equivalence relationships among the descriptors in the thesaurus. This is an advantage because the model may be used with almost any thesaurus and without having

K. Mellouli (Ed.): ECSQARU 2007, LNAI 4724, pp. 865–877, 2007.
© Springer-Verlag Berlin Heidelberg 2007

preclassified documents (in a large hierarchy, the amount of preclassified document necessary for training may be huge). On the other hand, this is also a weakness because any kind of information not considered in the thesaurus (e.g. other synonymy relations, specific information handled by documentalists,...) will not be taken into account. Consequently, we cannot expect very high rates of success in comparison with classifiers that are built starting from training data [3,5,9,13]. In this sense our proposal is more similar to the work in [1,2], where a method to populate an initially empty taxonomy is proposed. The working hypothesis is that a documentalist would prefer to confirm or discard a given classification hypothesis proposed by the system rather than examining all the possible alternatives.

Another important characteristic of our model is that is based on Bayesian networks. To the best of our knowledge, no Bayesian network-based models other than naive Bayes have been proposed to deal with this kind of problems [7]. We create a Bayesian network to model the hierarchical and equivalence relationships in the thesaurus. Then, given a document to classify, its terms are instantiated in the network and a probabilistic inference algorithm computes the posterior probabilities of the descriptors in the thesaurus.

In Section 2 we describe the proposed Bayesian network model of a thesaurus. The experimental evaluation is explained in Section 3. Finally, Section 4 contains the final remarks and some proposals for future work.

2 The Bayesian Network Model of a Thesaurus

In this section we shall first describe the general structure of a thesaurus and next the basic Bayesian network model that we propose to represent it, including the graphical structure, the conditional probabilities, the inference mechanism and some implementation details, and later a possible improvement. We assume that the reader has at least a basic background on Bayesian networks [12].

2.1 Thesaurus Structure

Any thesaurus comprises *descriptors* or *indexing terms*, *non-descriptors* or *entry terms* and *semantic relationships*, which may be equivalence, hierarchical and associative relationships. Descriptors are words or expressions which denote in unambiguous fashion the constituent concepts of the field covered by the thesaurus, whereas non-descriptors are words or expressions which in natural language denote the same or a more or less equivalent concept as a descriptor in the language of the thesaurus.

The *equivalence relationship* between descriptors and non-descriptors in fact covers relationships of several types: genuine synonymy, near-synonymy, antonymy and inclusion, when a descriptor embraces one or more specific concepts which are given the status of non-descriptors because they are not often used. It is usually represented by the abbreviations "UF" (Used For), between the descriptor and the non-descriptor(s) it represents, and "USE" between a non-descriptor

and the descriptor which takes its place. The *hierarchical relationship* between descriptors is shown by the abbreviations: "NT" (Narrower Term) between a generic descriptor and a more specific descriptor, and "BT" (Broader Term) between a specific descriptor and a more generic descriptor. Descriptors which do not contain other more specific descriptors are called *basic descriptors*; otherwise they are called *complex descriptors*. Descriptors which are not contained in any other broader descriptors are *top descriptors*. Sometimes a few descriptors may be polyhierarchical (they have more than one broader descriptor). This means that the hierarchical relationships do not form a tree but a graph. The *associative relationship*, shown by the abbreviation "RT" (Related Term), relates two descriptors that do not meet the criteria for an equivalence nor a hierarchical relationship. It is used to suggest another descriptor that would be helpful for the thesaurus user to search by. In this work we shall not consider associative relationships.

Example. Eurovoc is a multilingual thesaurus covering the fields in which the European Communities are active. Figure 1 displays the BT relationships between some descriptors of Eurovoc and the USE relationships between the non-descriptors and these descriptors. There are two complex descriptors, *abortion* and *birth control*, and four basic descriptors, *illegal abortion*, *therapeutic abortion*, *contraception* and *sterilisation*. The associated non-descriptors are: *legal abortion*, *termination of pregnancy* and *voluntary termination of pregnancy* for *abortion*; *birth spacing* for *birth control*; and *tubal ligation* and *vasectomy* for *sterilisation*.

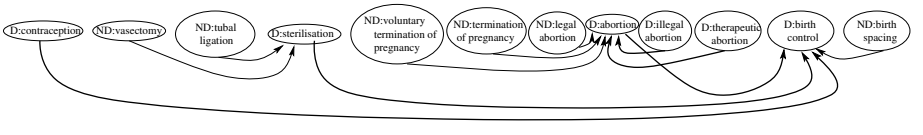

Fig. 1. BT (bold lines) and USE (normal lines) relationships for the descriptors and non-descriptors in the example about *abortion*

2.2 Basic Network Structure

In order to develop a Bayesian network (BN) for modeling a thesaurus, a naive approach would be to use a type of representation as the one in Fig. 1, containing descriptor and non-descriptor nodes, then adding term nodes representing the words in the thesaurus and connecting them with the descriptor and non-descriptor nodes that contain these words. This would result in a network structure as the one displayed in Fig. 2. The problem with this type of topology is that each descriptor node receives two or three kinds of arcs with different meaning (those from its non-descriptor nodes and those from its term nodes in the case of basic descriptor nodes and, for the case of complex descriptor nodes, also those arcs from the narrower descriptor nodes that they contain). As this would make

much more difficult the process of assigning the associated conditional probability distributions to the nodes, we propose a different topology. The idea is to distinguish between a concept and the descriptor and non-descriptors used to represent it.

Each concept, labeled identically as the descriptor representing it, will be a node in the network. We shall also distinguish between basic and complex concepts: the former do not contain other concepts, whereas the later are composed of other concepts (either basic or complex). Each descriptor and each non-descriptor in the thesaurus will also be nodes in the network. All the words or terms appearing in either a descriptor or a non-descriptor will be term nodes.

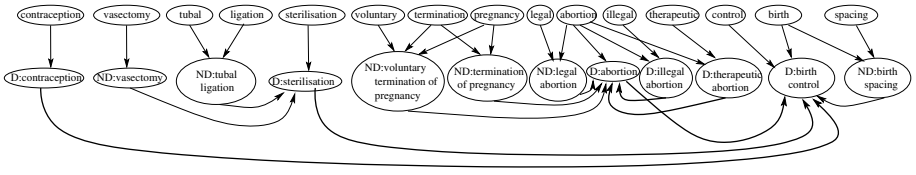

Fig. 2. Preliminary Bayesian network in the example about *abortion*

There is an arc from each term node to each descriptor and/or non-descriptor node containing it. There are also arcs from each non-descriptor node to the associated concept node (these arcs correspond with the USE relationships), as well as from the descriptor node representing the concept to the concept node itself.

As the complex concepts, in addition to its own specific information (descriptors and non-descriptors), are also containers of other concepts, for each complex concept we shall also create a duplicate (virtual) descriptor node which will receive the influence of the concepts contained in the complex concept. Therefore, there is an arc from each concept node which is not associated with a top descriptor to the virtual descriptor node associated with the broader complex concept(s) containing it (these arcs correspond with the BT relationships), as well as an arc going from each virtual descriptor node to its corresponding complex concept node.

We shall denote $\mathcal{T}$ the set of term nodes, $\mathcal{DE}$ and $\mathcal{ND}$ the sets of descriptor and non-descriptor nodes, respectively, $\mathcal{C}$ the set of concept nodes and $\mathcal{V}$ the set of virtual descriptor nodes. All the nodes will represent binary random variables. The domain of each variable is: $\{t^+, t^-\}\ \forall T \in \mathcal{T}$; $\{de^+, de^-\}\ \forall DE \in \mathcal{DE}$; $\{nd^+, nd^-\}\ \forall ND \in \mathcal{ND}$; $\{c^+, c^-\}\ \forall C \in \mathcal{C}$; $\{v^+, v^-\}\ \forall V \in \mathcal{V}$. For term nodes, their values indicate whether the term appear in the document to be classified. For descriptor and non-descriptor nodes, the values represent whether the corresponding descriptor or non-descriptor may be associated with the document. For concept nodes the values mean whether the concept is appropriate/relevant to classify the document. $Pa(X)$ will represent the parent set of a node X in the graph. The network topology that we are proposing is completely determined by specifying the parent set of each node: for each term node $T \in \mathcal{T}$, $Pa(T)$

is the empty set; for each descriptor and non-descriptor node $DE \in \mathcal{DE}$ and $ND \in \mathcal{ND}$, $Pa(DE)$ and $Pa(ND)$ are in both cases the set of term nodes associated with the words that appear in DE and ND, respectively; for each concept node $C \in \mathcal{C}$, $Pa(C)$ is the set of descriptor and non-descriptor nodes that define the concept and, in the case of complex concept nodes, also its associated virtual descriptor node, V_C; finally, for each virtual descriptor node $V \in \mathcal{V}$, $Pa(V)$ is the set of concept nodes (either basic or complex) contained in the corresponding complex concept.

For the previous example the corresponding subnetwork is shown in Fig. 3. The nodes labeled with D and ND are descriptor and non-descriptor nodes, respectively. The nodes labeled with C are concept nodes and those labeled with V are virtual descriptor nodes. The remaining nodes are term nodes.

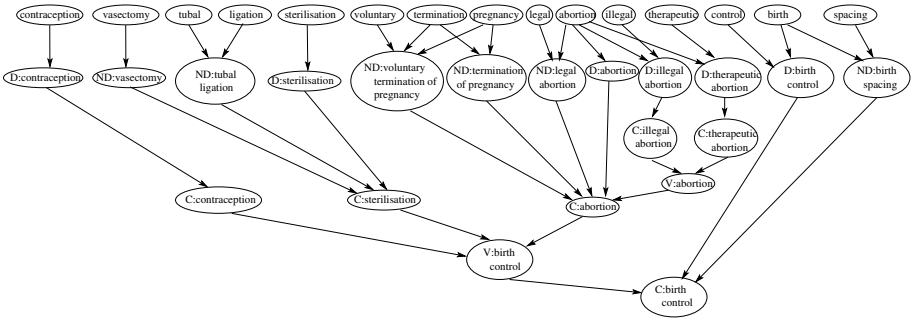

Fig. 3. Bayesian network in the example about *abortion*

2.3 Types of Conditional Probability Distributions

The probability distributions that must be specified are the prior probabilities for term nodes, $p(t^+)$, and the following conditional probabilities: for descriptor and non-descriptor nodes, $p(de^+|pa(DE))$ and $p(nd^+|pa(ND))$ respectively, for concept nodes, $p(c^+|pa(C))$, and for virtual descriptor nodes, $p(v^+|pa(V))$. In all the cases $pa(X)$ represents a configuration of the parent set $Pa(X)$ of the node X.

For the prior probabilities of term nodes we propose using a constant value, $p(t^+) = p_0$, $\forall T \in \mathcal{T}$ (although we shall see later that this is not an important issue at all).

As the treatment of the descriptor and non-descriptor nodes will the same, in order to simplify the exposition, from now on we will denote $\mathcal{D} = \mathcal{DE} \cup \mathcal{ND}$ and we will refer to both descriptor and non-descriptor nodes as descriptor nodes. An element in $\mathcal{D}$ will be denoted as D. For the conditional probabilities of a descriptor node D given the terms that it contains, $p(d^+|pa(D))$, we propose using a canonical additive model [4], employed in the information retrieval field:

$$\forall D \in \mathcal{D},\ p(d^+|pa(D)) = \sum_{T \in R(pa(D))} w(T, D)\ , \tag{1}$$

where $w(T, D)$ is the weight associated to each term T belonging to the descriptor D. $R(pa(D))$ is the subset of parents of D which are observed in the configuration $pa(D)$, i.e., $R(pa(D)) = \{T \in Pa(D) \,|\, t^+ \in pa(D)\}$. So, the more parents of D are observed the greater its probability of relevance. These weights can be defined in any way, the only restrictions are that $w(T, D) \geq 0$ and $\sum_{T \in Pa(D)} w(T, D) \leq 1$.

For the conditional probabilities of each concept node C given the descriptor nodes that define the concept and its virtual descriptor node (in the case of complex concept nodes), $p(c^+|pa(C))$, it is not appropriate to use the previous additive model, because each descriptor alone is supposed to be able to represent the concept, and this behaviour cannot be obtained using an additive model. So, we propose to use another kind of canonical model, namely an OR gate [12]:

$$\forall C \in \mathcal{C},\ p(c^+|pa(C)) = 1 - \prod_{D \in R(pa(C))} (1 - w(D, C)) \quad . \tag{2}$$

$R(pa(C)) = \{D \in Pa(C) \,|\, d^+ \in pa(C)\}$ and $w(D, C)$ is the probability that the descriptor D alone (the other descriptors being non relevant) makes concept C relevant, with $0 \leq w(D, C) \leq 1$.

For the conditional probabilities of each virtual descriptor node V given the concept nodes it comprises, $p(v^+|pa(V))$, we can use again the previous additive canonical model, because the more relevant are all the concepts contained in the complex concept associated to V, the more clearly this broader concept is appropriate:

$$\forall V \in \mathcal{V},\ p(v^+|pa(V)) = \sum_{C \in R(pa(V))} w(C, V) \quad . \tag{3}$$

$R(pa(V)) = \{C \in Pa(V) \,|\, c^+ \in pa(V)\}$ and $w(C, V)$ is the weight of the concept C in V, with $w(C, V) \geq 0$ and $\sum_{C \in Pa(V)} w(C, V) \leq 1$.

2.4 Quantifying the Conditional Probabilities

To define the weight of a term in a descriptor, $w(T, D)$, we propose a normalized tf-idf scheme:

$$w(T, D) = \frac{tf(T, D) * idf(T)}{\sum_{T' \in Pa(D)} tf(T', D) * idf(T')} \quad .$$

The *inverse descriptor frequency* of a term, $idf(T)$, is

$$idf(T) = \ln\left(\frac{m}{n(T)}\right) \quad ,$$

where $n(T)$ is the number of descriptors and non-descriptors in the thesaurus that contain the term T and m is the total number of descriptors and non-descriptors. The *term frequency* of a term in a descriptor, $tf(T, D)$, is the number of times that this term appears in the descriptor (which will be almost always equal to 1, because the descriptors usually contain very few words).

For the weights of the descriptors in the concepts, $w(D, C)$, a reasonable choice is a value near 1.0, because any descriptor associated with a concept represents it perfectly (descriptors and non-descriptors associated with a concept are assumed to be synonymous in the language of the thesaurus). In the experiments we have used $w(D, C) = 0.9$, in order to discriminate between concepts having a different number of descriptors that match with the document to be classified.

Finally, for the weights of the component concepts in each virtual descriptor, $w(C, V)$, we propose to use uniform weights (there is no reason to believe that a concept is more important than another one with respect to the broader concept containing them). Therefore:

$$w(C, V) = \frac{1}{|Pa(V)|} \ .$$

2.5 Inference

Given a document Q to be classified/indexed, the process is first to instantiate in the network the term nodes corresponding to the words appearing in Q as observed and the remaining term nodes as not observed. Let q be such a configuration of the term nodes in $\mathcal{T}$. Then we propagate this information through the network and compute the posterior probabilities of the concept nodes, $p(c^+|q)$. Finally, the descriptors associated with the concept nodes having greater posterior probability are used to classify the document.

To compute the posterior probabilities of the concept nodes, we can take advantage of both the network topology and the canonical models being considered. As all the term nodes are instantiated to either observed or non-observed, then all the descriptor nodes which are parents of a concept (including the associated virtual descriptor if it exists) are conditionally independent given q. In this case, taking into account that the canonical model for the concept nodes is an OR gate, we can compute these probabilities as follows [12]:

$$p(c^+|q) = 1 - \prod_{D \in Pa(C)} \left(1 - w(D, C)p(d^+|q)\right) \ .$$

As the weights $w(D, C)$ are all equal to 0.9, we have:

$$p(c^+|q) = 1 - \prod_{D \in Pa(C)} \left(1 - 0.9p(d^+|q)\right) \ . \tag{4}$$

The probabilities of the (non virtual) descriptor nodes can be calculated, according to the additive model being used, as follows [4]:

$$p(d^+|q) = \sum_{T \in Pa(D)} w(T, D)p(t^+|q) \ .$$

As $p(t^+|q) = 1 \ \forall T \in Pa(D) \cap Q$ and $p(t^+|q) = 0 \ \forall T \in Pa(D) \setminus Q$, we obtain:

$$p(d^+|q) = \sum_{T \in Pa(D) \cap Q} w(T, D) \ . \tag{5}$$

The computation of the posterior probabilities of the virtual descriptor nodes is also very simple, using again the properties of the additive canonical model considered:

$$p(v^+|q) = \frac{1}{|Pa(V)|} \sum_{C \in Pa(V)} p(c^+|q) \ . \tag{6}$$

This computation can be carried out as soon as the posterior probabilities of all the concept nodes included in V are known.

Therefore, we compute first the posterior probabilities of all the descriptor nodes using (5), then the posterior probabilities of the basic concept nodes (which have no virtual descriptor) using (4). Next, we can compute in a top-down manner the posterior probabilities of the virtual descriptor nodes and the complex concept nodes using (6) and (4), respectively.

2.6 Implementing the Model

In this section we shall study in more detail how to implement in an efficient way the proposed model. We start from the term nodes associated with the words appearing in the document to be classified. For each one of them, we accumulate the weights of these term nodes in the descriptor nodes containing them. After this process, each visited descriptor node D contains the value $v[D] = \sum_{T \in Pa(D) \cap Q} w(T, D)$, i.e. $p(d^+|q)$, according to (5) (the posterior probability of the non visited descriptor nodes is equal to zero).

Next, starting from each of the visited descriptor nodes, we visit the concept node containing it and compute the product $\prod_{D \in Pa(C)} (1 - 0.9v[D])$ progressively. After this step each visited basic concept node contains, according to (4), the value $v[C] = 1 - p(c^+|q)$ (the non visited basic concept nodes have a posterior probability equal to zero) and each visited complex concept node contains the value $v[C] = (1 - p(c^+|q))/(1 - 0.9p(v_c^+|q))$, because the contribution of its virtual descriptor node has not been computed yet.

Finally, we traverse the subgraph induced by the set of visited concept nodes and their descendants in a topological ordering (parents before children). If the visited node is a basic concept node C, we directly compute $p(c^+|q)$ (by setting $v[C] = 1 - v[C]$). If the visited node is a virtual node V, we compute its probability by adding the values already computed for its parent concept nodes and dividing by the number of parents, according to (6). If the visited node is a complex concept node C, we compute its probability by subtracting from 1 the value obtained by multiplying its stored value and the value already computed for its associated virtual node, $v[C] = 1 - v[C](1 - 0.9v[V_C])$. It can be easily seen that the complexity of this process is linear in the number of arcs in the graph[1]. It is worth mentioning that the actual implementation manages the BN implicitly, i.e. the Bayesian network is never explicitly constructed; instead, we directly use the BT, NT and USE relationships in the thesaurus, augmented

[1] More precisely, the complexity is linear in the number of arcs of the subgraph induced by the term nodes appearing in the document Q and their descendant nodes.

with two inverted file-like structures to store, for each word in the thesaurus, the lists of descriptors and non-descriptors that contain it.

2.7 Taking Degree of Coverage into Account

There is another dimension of the concepts in a thesaurus with respect to the document to be classified that we have not considered yet. We call this property the *coverage* of a concept with respect to a document, which tries to discriminate between concepts which are almost surely relevant for the document: if two concepts are initially considered equally relevant to classify a document but one of them includes more descriptors appearing in the document than the other, the former should be preferable. This strategy is motivated by the common guidelines being used to manually classify documents: we should use the most specific concepts available to bring out the main focus of a document and, if the document covers several specific concepts, then we should use as many specific concepts from different subtrees as required by the content of the document. *However, when several specific concepts are needed that fall within the same subtree structure, the broader concept should be assigned instead.*

Using the previous Bayesian network model, if, for instance, the three concepts which are included into a broader concept are completely relevant for a given document, then this broader concept also becomes completely relevant and therefore the four concepts would be (wrongly) assigned to the document.

To overcome this problem, we shall define the coverage of a concept C, $cov(C)$, as the set of concepts which are ancestors of C in the Bayesian network, together with C itself, i.e. all the concepts which are specializations (at different levels of granularity) of C. For example, the coverage of the concept *birth control* are the concepts *abortion*, *contraception*, *sterilisation*, *illegal abortion*, *therapeutic abortion* and *birth control*. Roughly speaking, the degree of coverage of a concept with respect to a document is the proportion of the document which is within the coverage of the concept. More concretely, $\forall C \in \mathcal{C}$, let us define $An_t(C) = \{T \in \mathcal{T} \,|\, \exists B \in cov(C), \exists D \in Pa(B) \text{ and } T \in Pa(D)\}$. In words, $An_t(C)$ is the set of terms in the thesaurus which are part of a descriptor associated to a concept in the coverage of C. We formally define the degree of coverage of a concept C with respect to a document Q, $dc(C, Q)$, as:

$$dc(C, Q) = \frac{\sum_{T \in An_t(C) \cap Q} idf(T)}{\sum_{T \in Q} idf(T)} \quad .$$

The decision about what descriptors to assign to a document should be made, not only depending on the probability of relevance of the concepts but also in terms of the degree of coverage of these concepts.

In order to formally include these ideas in the model, we shall think in terms of Decision theory, by defining a utility function based on the degree of coverage and then computing the expected utility of assigning a concept to a document. Those concepts having higher expected utility will be used to classify the document. If we define the utility of assigning the concept C to the document Q when

C is truly relevant as $dc(C, Q)$, and the utility of assigning C to Q when C is not relevant as zero, then the expected utility of assigning C to Q is simply $p(c^+|q) \times dc(C, Q)$.

3 Experimental Evaluation

Our experiments have been carried out using a data base provided by the Parliament of Andalucía at Spain, containing 7933 parliamentary initiatives manually classified using descriptors from an adapted version of the Eurovoc thesaurus. This version contains 5080 descriptors, 6975 non-descriptors and 7120 distinct words (excluding stopwords)[2]. The average number of assigned descriptors per initiative is 3.8. We have not used the full text of the initiatives but only a short summary (typically two or three lines of text). As our aim is not a complete but only a partial automation of the classification process, the selected performance measures have been the *average recall-precision curve* and the *average 11-point precision*[3], which are frequently used for category-ranking classifiers [14].

We have experimented with two alternatives: (1) the basic Bayesian network alone (BN) and (2) using coverage (BN+C). Moreover, each of these options has been tested with and without using stemming, although we always use stopword removal. The recall-precision curves of the four alternatives are displayed in Fig. 4, whereas the average 11-point precision values are shown in Table 1. With respect to the efficiency of the inference process, all the 7933 initiatives were classified in around 10 seconds on a computer equipped with an Intel Core2 duo 2GHz processor.

In order to assess the quality of the proposed BN-based models, we have also experimentally compared them with two simple benchmark methods. The first one [8,15] ranks concepts for a document based on word matching between the document and the lexical information associated to the concepts in the thesaurus, using a conventional vector space model (VSM) and the cosine measure: each document to be classified is considered as a query against a "document collection" where each "document", representing a concept, is indexed using the words appearing in the descriptor and non-descriptors which are associated with the concept. This approach uses only the lexical information, while topological (hierarchical) information is neglected. A second approach which also exploits the hierarchical information (HVSM) [1,2] is based on the idea that the meaning of a concept in the thesaurus is a specialization of the meaning of the broader concepts containing it[4]. Therefore, all the words appearing in the descriptors and non-descriptors of the broader concepts of a given concept are also used to index the "document" associated with this concept. The results obtained by

[2] The BN representing the thesaurus contains more than 25000 nodes.

[3] The precision values are interpolated at 11 points at which the recall values are 0.0, 0.1,. . ., 1.0, and then averaged.

[4] In the language of our Bayesian network model, these broader concepts would be the descendants of the concept being considered.

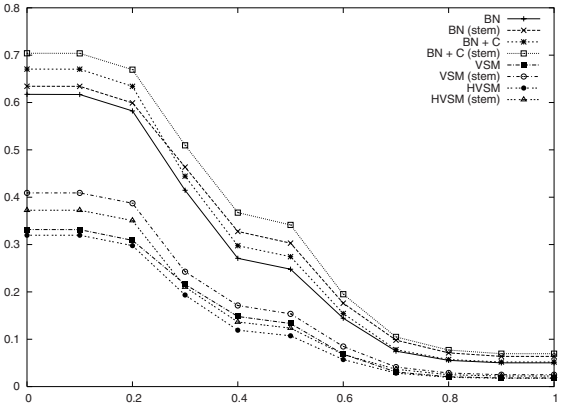

Fig. 4. Average recall-precision curves

Table 1. Average 11-point precision for the different experiments

Using stemming				Without using stemming			
BN+s	BN+C+s	VSM+s	HVSM+s	BN	BN+C	VSM	HVSM
0.3123	0.3466	0.1798	0.1582	0.2841	0.3123	0.1478	0.1361

these two benchmark models, once again with and without using stemming, are also displayed in Fig. 4 and Table 1.

Several conclusions may be obtained from these experiments: first, as the BN-based models always provide much better results than both the simple and hierarchical vector space models, it seems that the Bayesian network approach is useful in this classification problem. Second, stemming is also recommendable in this context, because its use always improves the results. Third, using coverage is clearly advantageous. Fourth, concerning the vector space model, in this case the use of the hierarchical information is self-defeating and produces results worse than those of the simple VSM[5]. Finally, the model performance is in general quite acceptable, specially at lower points of recall, reaching a precision near 70%.

4 Concluding Remarks

We have developed a Bayesian network-based model for hierarchical classification of documents from a thesaurus. The experimental results obtained using a large set of parliamentary initiatives from the Parliament of Andalucía and the Eurovoc thesaurus are encouraging, specially if we consider that no training data

[5] This contrasts with the results obtained in [1] in the context of hierarchical classification of documents into web directories, where the hierarchical VSM generally outperformed the simple VSM.

are used to build the model, and outperform those of the two simple benchmark methods considered.

For future research, we are planning to improve the model in three different ways: first, by considering the *context* of the terms/descriptors appearing in a document. The idea is to avoid assigning to a document a descriptor whose appearance may be incidental or their meaning within the document being quite different from the intended meaning within the thesaurus. Second, by taking also into account the associative relationships between descriptors in the thesaurus. Third, by integrating the model within a more general scheme where training data, in the form of preclassified documents, may also be used.

Acknowledgments

This work has been supported by the Spanish 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía' and 'Ministerio de Educación y Ciencia', under Projects TIC-276 and TIN2005-02516 respectively.

References

1. Adami, G., Avesani, P., Sona, D.: Clustering documents in a web directory. In: Proceedings of Fifth ACM Int. Workshop on Web Information and Data Management, pp. 66–73. ACM Press, New York (2003)
2. Adami, G., Avesani, P., Sona, D.: Clustering documents into a web directory for bootstrapping a supervised classification. Data & Knowledge Engineering 54, 301–325 (2006)
3. Chakrabarti, S., Dom, B., Agrawal, R., Raghavan, P.: Using taxonomy, discriminants, and signatures for navigating in text databases. In: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 446–455 (1997)
4. de Campos, L.M., Fernández-Luna, J.M, Huete, J.F.: The BNR model: foundations and performance of a Bayesian network-based retrieval model. International Journal of Approximate Reasoning 34, 265–285 (2003)
5. Dumais, S., Chen, H.: Hierarchical classification of web document. In: Proceedings of the 23th ACM International Conference on Research and Development in Information Retrieval, pp. 256–263. ACM Press, New York (2000)
6. Golub, K.: Automated subject classification of textual web documents. Journal of Documentation 62(3), 350–371 (2006)
7. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In: Proceedings of the 14th International Conference on Machine Learning, pp. 170–178 (1997)
8. Larson, R.R.: Experiments in automatic library of congress classification. Journal of the American Society for Information Science 43(2), 130–148 (1992)
9. Lauser, B., Hotho, A.: Automatic multi-label subject indexing in a multilingual envirinment. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 140–151. Springer, Heidelberg (2003)
10. Medelyan, O., Witten, I.: Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries, pp. 296–297 (2006)

11. Moskovitch, R., Cohen-Kashi, S., Dror, U., Levy, I.: Multiple hierarchical classification of free-text clinical guidelines. Artificial Intelligence in Medicine 37(3), 177–190 (2006)
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan and Kaufmann, San Mateo (1988)
13. Ruiz, M., Srinivasan, P.: Hierarchical text categorization using neural networks. Information Retrieval 5(1), 87–118 (2002)
14. Sebastiani, F.: Machine Learning in automated text categorization. ACM Computing Surveys 34, 1–47 (2002)
15. Yang, Y.: An evaluation of statistical approaches to text categorization. Information Retrieval 1, 69–90 (1999)