International Journal of Approximate Reasoning 50 (2009) 932–944

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Bayesian network models for hierarchical text classification from a thesaurus

Luis M. de Campos, Alfonso E. Romero *

Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación, Universidad de Granada, Daniel Saucedo Aranda, s/n, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 18 March 2008
Received in revised form 12 September 2008
Accepted 28 October 2008
Available online 27 November 2008

Keywords:
Bayesian networks
Document categorization
Hierarchical classification
Thesauri

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). Our model can be used without having preclassified training documents, although it improves its performance as long as more training data become available. We have tested the classification model using a document dataset containing parliamentary resolutions from the regional Parliament of Andalucía at Spain, which were manually indexed from the Eurovoc thesaurus, also carrying out an experimental comparison with other standard text classifiers.

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 resolutions, the Food and Agricultural Organization employs Agrovoc to categorize its documents, several organizations use the National Agriculture Library Thesaurus (NALT), and the National Library of Medicine 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. We believe that it is not realistic to try to design a completely automatic classification process, given the critical nature of the classification task in many contexts, and final human supervision will always be required in real environments.

The scope of our research is therefore automatic subject indexing from a controlled vocabulary [8,17] and hierarchical text classification [18,21]. There are several characteristics in this problem which make it difficult: (1) as each descriptor in the thesaurus represents a different class/category, it is a problem of high dimensionality (we are managing several thousand descriptors); (2) it is also a multi-label problem, because a document may be associated with several classes, exhibiting also a high variability in the number of descriptors being assigned to each document[1]; (3) there are explicit (hierarchical) relationships between the class labels, so that they are not independent among each other; (4) the training data can be quite unbalanced, having a very different number of documents associated to each class.

* Corresponding author.
E-mail addresses: lci@decsai.ugr.es (L.M. de Campos), aeromero@decsai.ugr.es (A.E. Romero).

[1] Between 1 and 14 in the document collection used in the experiments.

0888-613X/$ - see front matter Crown Copyright © 2008 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2008.10.006

An important characteristic of the model that we are going to propose is that no training is required to start using the system. Initially we shall exploit only the hierarchical and lexical information from the thesaurus to build the classifier. This is an advantage because the model may be used with almost any thesaurus and without having preclassified documents (in a large hierarchy, the amount of preclassified documents 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 relations, specific information handled by documentalists,...) will not be taken into account and, therefore, we should not expect very high success rates in comparison with classifiers that are built using training data [4,7,14,20]. In this sense our initial 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.

Nevertheless, the proposed model can also naturally incorporate training data in order to improve its performance: The information provided by preclassified documents can be appropriately merged with the hierarchical and equivalence relationships among the descriptors in the thesaurus, in order to obtain a classifier better than the one we would obtain by using only the training documents.

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 [12]. We create a Bayesian network to model the hierarchical and equivalence relationships in the thesaurus, and next we extend it to also use training data. Then, given a document to be classified, its terms are instantiated in the network and a probabilistic inference algorithm, especifically designed and particularly efficient, computes the posterior probabilities of the descriptors in the thesaurus.

The paper is organized as follows: In Section 2 we describe the proposed Bayesian network[2] model of a thesaurus, whereas the extension of the model to cope with training data is described in Section 3. The experimental evaluation is explained in Section 4. Finally, Section 5 contains the final remarks and some proposals for future work.

2. The Bayesian network representing a thesaurus

In this section we shall first introduce basic notions relative to the composition and structure of a thesaurus; next, we describe the Bayesian network model proposed to represent it, including the graphical structure, the conditional probabilities and the inference mechanism.

2.1. Thesauri

Broadly speaking, a thesaurus consists of a set of terms, which are relevant to a certain domain of knowledge, and a set of semantic relationships between them. The basic units of a thesaurus are *descriptors* or *indexing terms*, which are words or expressions which denote in unambiguous fashion the constituent concepts of the field covered by the thesaurus. A thesaurus also comprises *non-descriptors* or *entry terms*, which are words or expressions that denote the same or a more or less equivalent concept as a descriptor in the language of the thesaurus. The three most common types of semantic relationships are equivalence, hierarchical and associative relationships.

The *equivalence relationship* between descriptors and non-descriptors may cover 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: "BT" (Broader Term) between a specific descriptor and a more generic descriptor, and its dual "NT" (Narrower Term) between a generic descriptor and a more specific 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 are polyhierarchical (they have more than one broader descriptor), which means that the hierarchical relationships may form a graph instead of a tree. 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.

2.1.1. Example

Eurovoc is a multilingual thesaurus that provides a means of indexing the documents in the documentation systems of the European institutions and of their users. Fig. 1 displays the BT relationships between some descriptors of Eurovoc and the USE relationships between the non-descriptors and these descriptors.[3] There are two complex descriptors, *health service* and *health policy*, and three basic descriptors, *medical centre*, *medical institution* and *psychiatric institution*. *Health service* is the broad-

[2] We assume that the reader has at least a basic background on Bayesian networks.

[3] The English version of Eurovoc comprises 6645 descriptors and 6769 non-descriptors, together with 6669 BT/NT relationships.

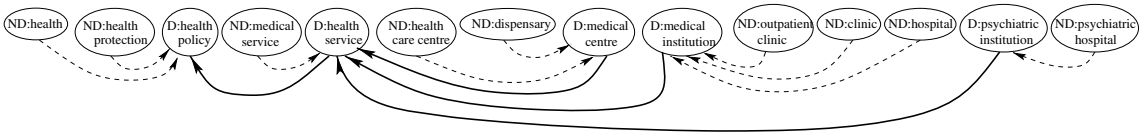

Fig. 1. BT (bold lines) and USE (dashed lines) relationships for the descriptors (D) and non-descriptors (ND) in the example about *health*.

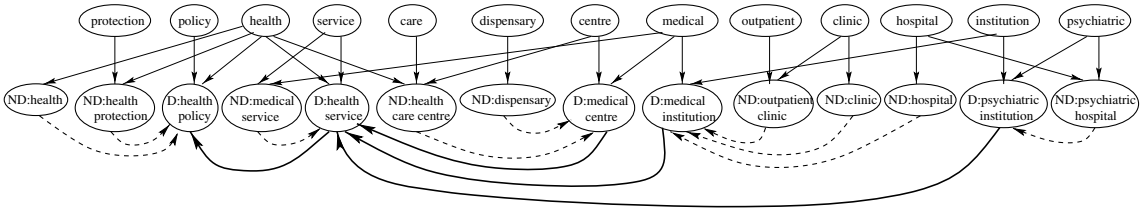

Fig. 2. Preliminary Bayesian network in the example about *health*.

er term of *medical centre*, *medical institution* and *psychiatric institution*; *health policy* is in turn the broader term of *health service* and also of other five descriptors which are not displayed.[4] The associated non-descriptors are: *medical service* for *health service*; *health* and *health protection* for *health policy*; *dispensary* and *health care centre* for *medical centre*; *clinic*, *hospital* and *outpatients' clinic* for *medical institution*; and *psychiatric hospital* for *psychiatric institution*.

2.2. Bayesian network structure

A simple approach for modeling a thesaurus as a Bayesian network would be to use a type of representation directly based on the graph displayed 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 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 key ideas are: (1) to explicitly distinguish between a concept and the descriptor and non-descriptors used to represent it and (2) to clearly separate, through the use of additional nodes, the different information sources (hierarchy and equivalence relationships) influencing on a concept.

According to the first key idea, each concept, labeled identically as the descriptor representing it, will be a node C 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 D and ND in the network. All the words or terms appearing in either a descriptor or a non-descriptor will be term nodes T. To accomplish with the second key idea, for each concept node C we shall also create two (virtual) nodes: E_C, which will receive the information provided by the equivalence relationships involving C; and H_C, which will collect the hierarchical information, i.e. the influence of the concepts contained in C.

With respect to the links, 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, associated to a concept node C, to the corresponding virtual node E_C (these arcs correspond with the USE relationships), as well as from the own descriptor node associated with the concept C to E_C. There is also an arc from each concept node C' (excluding those nodes which are associated with a top descriptor) to the virtual node(s) H_C associated with the broader complex concept(s) C containing C' (these arcs correspond with the BT relationships). Finally, there are arcs from the virtual nodes E_C and H_C to its associated concept node C, representing that the relevance of a given concept will directly depend on the information provided by the equivalence (E_C node) and the hierarchical (H_C node) relationships. For the previous example the corresponding subnetwork is shown in Fig. 3. It should be noticed that this model is slightly different and more general than the one proposed in [6], where virtual equivalence nodes were not considered.

We shall denote $\mathscr{T}$ the set of term nodes, $\mathscr{DE}$ and $\mathscr{ND}$ the sets of descriptor and non-descriptor nodes, respectively, $\mathscr{C}$ the set of concept nodes, and $\mathscr{E}$ and $\mathscr{H}$ the sets of virtual equivalence and hierarchical nodes, respectively. All the nodes will represent binary random variables. The domain of each variable is: $\{t^+, t^-\}\forall T \in \mathscr{T}; \{de^+, de^-\}\forall DE \in \mathscr{DE}; \{nd^+, nd^-\}\forall ND \in$

[4] These non displayed descriptors are *health care system*, *health costs*, *health expenditure*, *health statistics* and *organisation of health care*.

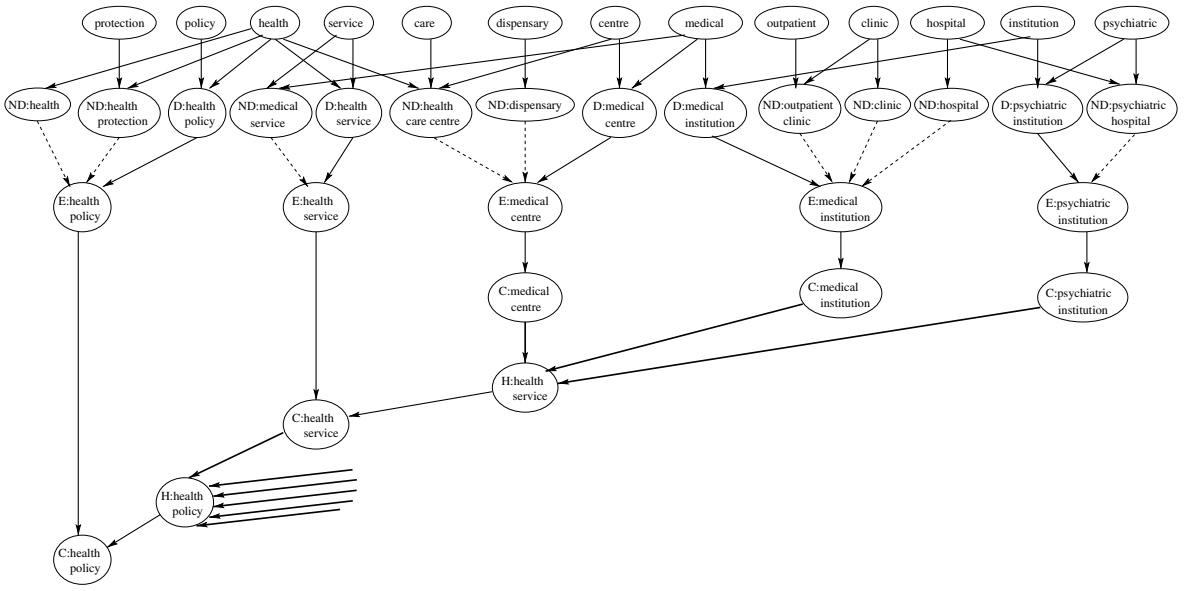

Fig. 3. Bayesian network in the example about *health*.

$\mathscr{N}\mathscr{D}; \{c^+, c^-\}\forall C \in \mathscr{C}; \{e^+, e^-\}\forall E \in \mathscr{E}; \{h^+, h^-\}\forall H \in \mathscr{H}$. 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 and their associated virtual 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 proposed network topology is completely determined by specifying the parent set of each node: for each term node $T \in \mathscr{T}, Pa(T)$ is the empty set; for each descriptor and non-descriptor node $DE \in \mathscr{D}\mathscr{E}$ and $ND \in \mathscr{N}\mathscr{D}, Pa(DE)$ 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 virtual equivalence node $E_C \in \mathscr{E}, Pa(E_C)$ is the set of descriptor and non-descriptor nodes that define the concept C; for each virtual hierarchical node $H_C \in \mathscr{H}, Pa(H_C)$ is the set of concept nodes contained in the corresponding complex concept C; finally, for each concept node $C \in \mathscr{C}, Pa(C) = \{E_C, H_C\}$, the set of its two associated virtual nodes.

2.3. Conditional probability distributions

The probability distributions that must be specified are the prior probabilities for term nodes, $p(t^+)$, and the conditional probabilities for the remaining nodes: $p(de^+|pa(DE))$, $p(nd^+|pa(ND))$, $p(c^+|pa(C))$, $p(e^+|pa(E))$ and $p(h^+|pa(H))$. 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, $p(t^+) = p_0, \forall T \in \mathscr{T}$ (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 shall denote $\mathscr{D} = \mathscr{D}\mathscr{E} \cup \mathscr{N}\mathscr{D}$ and we shall refer to both descriptor and non-descriptor nodes as descriptor nodes. An element in $\mathscr{D}$ 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 [5], which has been successfully employed in the information retrieval field and will allow us to perform exact inference efficiently:

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

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 $w(T,D) \geqslant 0$ and $\sum_{T \in Pa(D)} w(T, D) \leqslant 1$.

To define the weight of a term in a descriptor, $w(T,D)$, we propose a normalized tf–idf scheme, as those are frequently used in I.R.:

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

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

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

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 conditional probabilities of each virtual equivalence node E_C given the descriptor nodes that define the concept C, $p(e_c^+|pa(E_C))$, 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 [19]:

$$\forall E_C \in \mathscr{E}, p(e_c^+|pa(E_C)) = 1 - \prod_{D \in R(pa(E_C))} (1 - w(D,C)). \tag{2}$$

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

For the weights of the descriptors in the concepts, $w(D,C)$, a reasonable choice is a high 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).[5]

For the conditional probabilities of each virtual hierarchical node H_C given the concept nodes it comprises, $p(h_c^+|pa(H_C))$, we can use again the previous additive canonical model, because the more relevant are all the concepts contained in the complex concept C associated to H_C, the more clearly this broader concept is appropriate[6]:

$$\forall H_C \in \mathscr{H}, p(h_c^+|pa(H_C)) = \sum_{C' \in R(pa(H_C))} w(C', H_C). \tag{3}$$

$R(pa(H_C)) = \{C' \in Pa(H_C) | c' + \in pa(H_C)\}$ and $w(C',H_C)$ is the weight of the concept C' in H_C, with $w(C',H_C) \geqslant 0$ and $\sum_{C' \in Pa(H_C)} w(C', H_C) \leqslant 1$.

For these weights $w(C',H_C)$, we propose to use uniform weights (there is no prior reason to believe that a concept is more important than another one with respect to the broader concept containing them). Therefore:

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

Finally, for the conditional probabilities of each concept node given its associated virtual nodes, $p(c^+|\{e_c,h_c\})$, we again propose an OR gate (a concept may become relevant either because of its own lexical information (its descriptor and non-descriptors) or because most of the narrower concepts contained in it become relevant): $\forall C \in \mathscr{C}$,

$$p(c^+|\{e_c,h_c\}) = \begin{cases} 1 - (1 - w(E_C,C))(1 - w(H_C,C)) & \text{if } e_c = e_c^+, h_c = h_c^+, \\ w(E_C,C) & \text{if } e_c = e_c^+, h_c = h_c^-, \\ w(H_C,C) & \text{if } e_c = e_c^-, h_c = h_c^+, \\ 0 & \text{if } e_c = e_c^-, h_c = h_c^-, \end{cases} \tag{4}$$

where $w(E_C,C)$ and $w(H_C,C)$ are the weights or importance attributed to the equivalence and hierarchical information, respectively, with $0 \leqslant w(E_C,C) \leqslant 1$ and $0 \leqslant w(H_C,C) \leqslant 1$.

2.4. Inference

The procedure used to classify a given document Q would be as follows: first we instantiate in the network the term nodes corresponding to the words appearing in Q as observed and the remaining term nodes as not observed.[7] Let q be such a configuration of the term nodes in $\mathscr{T}$. Next, 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.

We can take advantage of both the network topology and the canonical models being considered in order to compute the posterior probabilities of the concept nodes. As all the term nodes are instantiated to either observed or non-observed, then

[5] In order to discriminate between concepts having a different number of descriptors that match with the document to be classified, it is preferable not to use a value equal to 1.0 (otherwise we cannot distinguish between a concept with only one relevant descriptor and other having several relevant descriptors).

[6] 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.*

[7] For that reason the values of the prior probabilities of the term nodes are not important.

all the descriptor nodes which are parents of a virtual equivalence node are conditionally independent given q. In the same way, the virtual nodes E_C and H_C associated to a concept node C are also conditionally independent given q. Therefore, taking into account that the canonical model for both virtual equivalence nodes and concept nodes is an OR gate, we can compute these probabilities as follows [19]:

$$p(e_c^+|q) = 1 - \prod_{D \in Pa(E_C)} (1 - w(D,C)p(d^+|q)). \tag{5}$$

$$p(c^+|q) = 1 - (1 - w(E_C,C)p(e_c^+|q))(1 - w(H_C,C)p(h_c^+|q)). \tag{6}$$

The probabilities of the descriptor nodes can be calculated, according to the properties of the additive model being used, as follows [5]:

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

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

$$p(d^+|q) = \sum_{T \in Pa(D) \cap Q} w(T,D). \tag{7}$$

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

$$p(h_c^+|q) = \frac{1}{|Pa(H_C)|} \sum_{C' \in Pa(H_C)} p(c'^+|q). \tag{8}$$

Therefore, we compute first the posterior probabilities of all the descriptor nodes using (7), then the posterior probabilities of the virtual equivalence nodes using (5). Next, we can compute in a top-down manner the posterior probabilities of the virtual hierarchical nodes and the concept nodes using (8) and (6), respectively.

Now, let us 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 $v[D] = \sum_{T \in Pa(D) \cap Q} w(T,D)$ (which coincides with $p(d^+|q)$, according to (7)). The posterior probabilities of the non visited descriptor nodes are equal to zero.

Next, starting from each of the visited descriptor nodes, we would visit the virtual equivalence node containing it and compute progressively the product $\prod_{D \in Pa(E_C)} (1 - w(D,C)\, v[D])$. After this step each visited virtual equivalence node contains, according to (5), the value $v[E_C] = 1 - p(e_c^+|q)$ (the non visited virtual equivalent nodes have a posterior probability equal to zero).

Finally, we traverse the subgraph induced by the set of visited virtual equivalence 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 $v[C] = w(E_C,C)(1 - v[E_C])$ (because there is no hierarchical information for basic concept nodes). If the visited node is a virtual hierarchical node H_C, we compute its probability by accumulating in $v[H_C]$ the values already computed for its parent concept nodes and dividing by the number of parents, according to (8). If the visited node is a complex concept node C, we compute its probability by setting $v[C] = 1 - (1 - w(E_C,C)v[E_C])(1 - w(H_C,C)v[H_C])$.

It can be easily seen that the complexity of this process is linear in the number of arcs in the graph or, more precisely, linear in the number of arcs of the subgraph induced by the term nodes appearing in the document Q and their descendant nodes. It is worth mentioning that in the actual implementation the Bayesian network is never explicitly constructed; instead, we directly use the BT, NT and USE relationships in the thesaurus, augmented with two inverted file-like structures to store, for each word in the thesaurus, the lists of descriptors and non-descriptors containing it.

3. Extending the model to cope with training information

The model proposed so far does not use training information, in the form of preclassified documents. However, it is quite simple to include this type of information into the Bayesian network model, thus obtaining a supervised classifier. Following with the previously used idea of clearly separating the different sources of information relative to each concept, then we will add a new parent node T_C, called virtual training node, to each concept node C (in addition to those virtual nodes H_C and E_C representing hierarchical and equivalence relationships), representing the information obtained for this concept from the training documents. In other words, this node T_C will contain the posterior probability distribution for the relevance of the concept, predicted by a (probabilistic) supervised classifier. This information will be merged with those obtained from hierarchy and equivalence through an OR gate.

Although, in principle, we could use any supervised classifier able to give a probability distribution as the output, we are going to propose a classifier which is particularly coherent with the thesaurus model, that we call the *OR gate Bayesian net-*

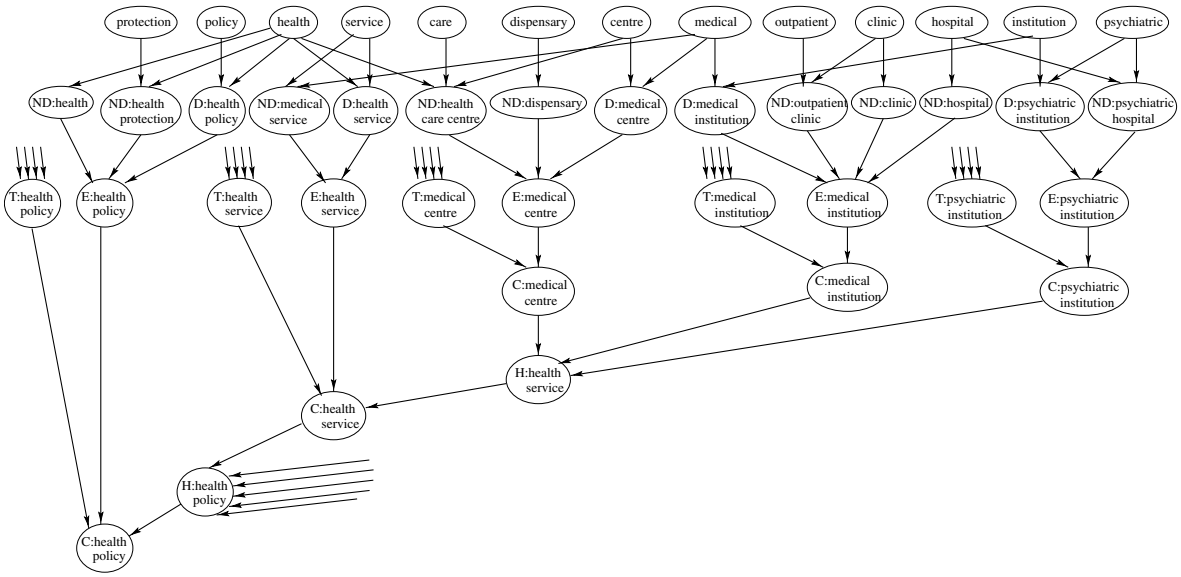

Fig. 4. Extended Bayesian network to include training information, in the example about *health*.

work classifier: The network structure is fixed, having an arc going from each term node T to the virtual training node T_C if this term appears in training documents which are associated with the concept C.[8] Fig. 4 displays the corresponding network for the example about health.

Concerning the numerical information, we include a new parameter $w(T_C,C),0 \leqslant w(T_C,C) \leqslant 1$, representing the contribution of the training information to the relevance of the concept C, so that the new conditional distribution of a concept node is

$$\forall C \in \mathscr{C}, p(c^+|pa(C)) = 1 - \prod_{X_C \in R(pa(C))} (1 - w(X_C, C)), \tag{9}$$

where X_C represents either E_C, H_C or T_C. The posterior probability of each concept given a document, $p(c^+|q)$, is therefore calculated as:

$$p(c^+|q) = 1 - (1 - w(E_C, C)p(e_c^+|q))(1 - w(H_C, C)p(h_c^+|q))(1 - w(T_C, C)p(t_c^+|q)). \tag{10}$$

The conditional distributions of the new virtual training nodes are defined, according to the OR gate model, using the weights $w(T,T_C)$ associated to each term T appearing in training documents which have been assigned to concept C:

$$\forall C \in \mathscr{C}, p(t_c^+|pa(T_C)) = 1 - \prod_{T \in R(pa(T_C))} (1 - w(T, T_C)). \tag{11}$$

The posterior probabilities can be computed as follows, using again the properties of the OR gate model:

$$p(t_c^+|q) = 1 - \prod_{T \in Pa(T_C) \cap Q} (1 - w(T, T_C)). \tag{12}$$

In order to take into account the number of times a word T occurs in a document Q,n_{TQ}, we replicate each node T n_{TQ} times, so that the posterior probabilities then become

$$p(t_c^+|q) = 1 - \prod_{T \in Pa(T_C) \cap Q} (1 - w(T, T_C))^{n_{TQ}}. \tag{13}$$

The estimation of the weights $w(T,T_C)$ can be done in several ways. The simplest one is to estimate $w(T,T_C)$ as $p(c^+|t^+)$, the estimated conditional probability of concept C given that the term T is present. We can do it by using the Laplace estimation:

$$w(T, T_C) = \frac{N_{TC} + 1}{N_T + 2}, \tag{14}$$

where N_{TC} is the number of times that the term T appears in documents associated to concept C and N_T is the number of times that the term T appears in all the training documents.

[8] Notice that these terms are no longer restricted to be part of the descriptors in the thesaurus, they are the terms found in the training documents.

Table 1
Performance measures for the experiments without using training documents (the two best values for each column are marked in boldface).

Models	Micro BEP	Macro BEP	Av. prec.	Micro F1@5	Macro F1@5
BN 0.8, 0	0.26244	0.20394	0.29967	0.30811	0.17661
BN 0.9, 0	**0.28241**	0.20234	0.30700	**0.31419**	0.18419
BN 0.8, 0.8	0.26068	**0.21208**	0.30500	0.30845	0.17521
BN 0.9, 0.9	**0.26881**	**0.20903**	**0.31321**	**0.31473**	**0.18433**
BN 0.9, 1.0	0.26636	0.20880	**0.31261**	0.31381	0.18265
BN 1.0, 1.0	0.25584	0.20768	0.27870	0.30963	**0.18865**
VSM	0.15127	0.18772	0.18061	0.20839	0.17016
HVSM	0.13326	0.17579	0.17151	0.20052	0.14587

4. Experimental evaluation

Our experiments have been carried out using a database provided by the Parliament of Andalucía at Spain, containing 7933 parliamentary resolutions 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). The BN representing the thesaurus would therefore contain more than 30,000 nodes. The average number of assigned descriptors per document is 3.8. We have not used the full text of the documents but only a short summary (typically two or three lines of text). In our experiments we always use stemming[9] and stopword removal.

The evaluation takes into account that our aim is not a complete but only a partial automation of the classification process, showing to the user an ordered list of the most probable descriptors.[10] Then, as performance measures, we have firstly selected the typical measure used in multi-label categorization problems: *breakeven* point (BEP).[11] This measure will be computed in microaverage and macroaverage. Two more measures tipically used in the Information Retrieval community will be used too: the F_1 *measure*[12] at fifth document level (the one obtained by assuming that the system assigns to each document the *five* most probable descriptors), and the *average 11-point precision*.[13] As in the case of the breakeven point, we shall compute the micro and macro averages of the F_1 measure. In all the measures, a higher value means a better performance of the model.

4.1. Experiments without using training documents

In order to assess the quality of the proposed model without using training data, we have also experimentally compared it with two simple benchmark methods. The first one [13,23] 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.[14] 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.

Several combinations of parameters have been tested for our Bayesian network-based model (BN). In particular, the parameters chosen to be variable have been the weights $w(H_C,C)$ and $w(D,C)$. As stated in subsection 2.3, we have chosen high values for the weight $w(D, C)$ (0.8 and 0.9), together with the value 1.0. In order to test the value of the hierarchical information, we have selected both high values (0.8, 0.9 and 1.0) and a low value (0.0). On the other hand, the value of the weight of the equivalence relationships $w(E_c, C)$ has been fixed to 1.0. Then, a value of, for example, "BN 0.9, 0.8" in the table of results, Table 1, means the Bayesian network model with $w(D,C) = 0.9$ and $w(H_C,C) = 0.8$.

The main conclusion that may be obtained from these experiments is that the Bayesian network approach is useful in this classification problem, since it always provides much better results than both the simple and hierarchical vector space models. The model performance is in general quite acceptable, taking into account that no training documents have been used. 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.[15]

[9] The spanish version of the well-known Porter algorithm, implemented in the Snowball package.
[10] We are therefore using an instance of the so-called category-ranking classifiers [21].
[11] The point where precision equals recall, by moving a threshold.
[12] The harmonic mean of precision and recall.
[13] The precision values are interpolated at 11 points at which the recall values are 0.0, 0.1, ..., 1.0, and then averaged.
[14] In the language of our Bayesian network model, these broader concepts would be the descendants of the concept being considered.
[15] 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.

Table 2
Performance measures for the experiments using training documents (the three best values for each column are marked in boldface).

Models	Micro BEP	Macro BEP	Av. prec.	Micro F1@5	Macro F1@5
Naive Bayes	0.39169	0.15529	0.62244	0.50310	0.20467
Rocchio	0.33921	0.35966	0.43732	0.40489	0.33921
OR gate	0.38671	0.43936	0.56236	0.41367	0.24629
SVM	**0.63069**	**0.48788**	0.69295	0.56393	**0.41040**
SBN 0.0, 0.9	0.54825	0.43361	0.66834	0.54066	0.33414
SBN 0.0, 0.8	0.55146	0.43257	0.67111	0.54194	0.33615
SBN 0.0, 0.5	0.55564	0.43126	0.67532	0.54491	0.33978
SBN 0.0, 0.1	0.55683	0.43115	0.67720	0.54572	0.34144
SBN 0.9, 0.0	0.55339	0.47328	0.66096	0.56891	0.36754
SBN 0.8, 0.0	0.57962	0.47986	0.68292	**0.58167**	**0.38591**
SBN 0.5, 0.0	**0.58353**	**0.48585**	**0.70221**	0.57801	0.37900
SBN 0.1, 0.0	0.56163	0.46114	0.68658	0.55296	0.35002
SBN 0.8, 0.1	0.57955	0.47940	0.68299	**0.58186**	**0.38629**
SBN 0.5, 0.1	**0.58307**	0.48564	**0.70245**	**0.57852**	0.38046
SBN 0.5, 0.5	0.58245	**0.48750**	**0.70140**	0.57835	0.37797
SBN 0.8, 0.8	0.56807	0.48157	0.67454	0.57537	0.37303
SBN 0.9, 0.9	0.54028	0.47345	0.65032	0.56289	0.35766
SBN 1.0, 1.0	0.49185	0.46066	0.59143	0.53281	0.32129

The parameters chosen show better performance for a weight of the descriptors near 1.0, than the 1.0 itself. Although the usage of a hierarchy weight distinct to 0.0 does not strongly boost the results, it performs little improvements in the measures, specially in the average precision.

With respect to the efficiency of the inference process, all the 7933 resolutions were classified in around 10 s on a computer equipped with an Intel Core2 duo 2 GHz processor.

4.2. Experiments using training documents

In this section we shall evaluate the results obtained by the model using training documents. We want also to evaluate how the system improves its performance as more training data are available. In all the computed measures through the experiments, we shall use 5-folds cross-validation over the same five partitions of the collection. The presented measures, then, will be the average of the five obtained values. The evaluation will be carried out with the same five measures chosen for the previous experimentation, in order to make both comparable.

In the first part of the experimentation, the supervised approach of our Bayesian network will be compared against four pure supervised approaches to multi-label classification. Concretely, we will use the multinomial naive Bayes model [15,16], the Rocchio classifier [9], support vector machines (SVM) [10] and a standalone OR gate classifier model, constructed using information only from training documents (and not taking into consideration neither the thesaurus lexical information nor its hierarchical structure).[16]

The first set of experiments, whose results are displayed in Table 2, compares the four supervised approaches with the model using training documents, tuning some parameters. As stated before, $w(D,C)$ should have a high value, near 1.0. This parameter will be fixed to 0.9 (a value which provides good results on the previous experimentation). On the other hand, the weight of the training information, $w(T_C,C)$, will be high, and also fixed (to 1.0 in this case). Therefore, the two free tunable parameters we will consider in the model will be the weight of the hierarchy, $w(H_C,C)$, and the weight of the equivalence relationships, $w(E_C,C)$. In Table 2, the supervised version of our Bayesian network model will be noted as "SBN a,b", where a will be the weight $w(E_c,C)$ and b will be $w(H_C,C)$. From a certain viewpoint, we want to study the contribution of these two sources of information (hierarchical and lexical) to the baseline model (the standalone OR gate classifier). This leads us to the two following questions. Does information from the terms of the thesaurus help in the supervised case? And the second one, does information from the hierarchical relationships of the thesaurus helps now?

These experiments show up that adding hierarchical information ("SBN 0.0, X") to the OR gate model clearly improves the classification results. Moreover, adding textual information ("SBN X, 0.0") without hierarchical information also boots classification results. In this case, the hierarchy added to the lexical information of the thesaurus does not make a significant advance, but it improves the results, being the "SBN 0.5, 0.1" and the "SBN 0.8, 0.1" two of the best performing configurations we have tested. The results in Table 2 show that our Bayesian network model systematically obtains better results than two classical supervised classifiers (Rocchio and naive Bayes) and one 'uninformed' version of itself (standalone OR gate), and even outperforms SVM in some cases.

Nevertheless, it should be noticed that in any case the performance measures obtained are not very high. This can be explained if we consider the fact that performance decreases as the number of categories in the problem being considered

[16] In all the cases we used our own implementations of these algorithms, except in the case of SVM, where the software package SVM Light [11] was used.

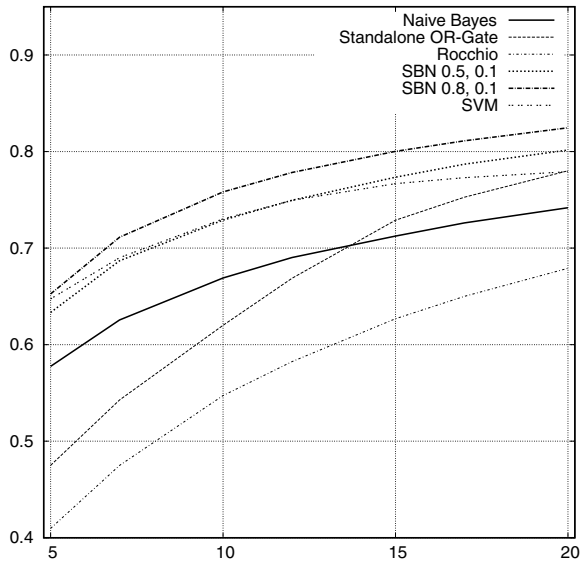

Fig. 5. Microaveraged recall values computed for incremental number of displayed categories.

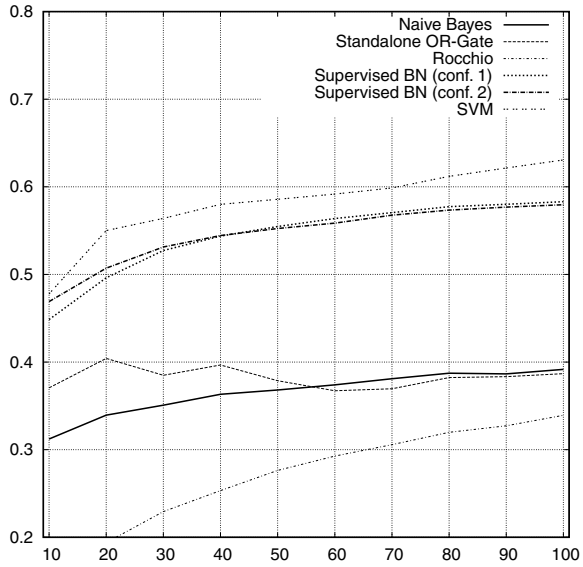

Fig. 6. Microaveraged breakeven point computed for incremental percentage of training data.

increases [3,22], and in our case the number of categories is quite high (in the order of thousands). However, as we explained earlier, our goal is not to replace the human experts but to help them, by providing an ordered list where the correct descriptors can be found in the first positions in the list. In order to show that this is indeed the case, we have carried out another experiment to compute the average recall values[17] obtained by the different classifiers when we display to the user the n top-ranked categories, for $n = 5$, 7, 10, 12, 15, 17 and 20. The results are displayed in Fig. 5.

We can observe in the figure that one of our models finds 65% of the true categories among the first five categories in the list, 75% among the first ten and 80% among the first fifteen (from a list of 5080 possible categories). We believe that any human indexer would consider useful a system having these characteristics.

The second part of the experimentation will test the classification models in an environment where not all the training data is available. In these experiments, for a same test partition, all the classifiers will be trained with the 10%, 20%,. . .,100%

[17] The proportion of correctly assigned categories with respect to the total number of true categories associated with each document.

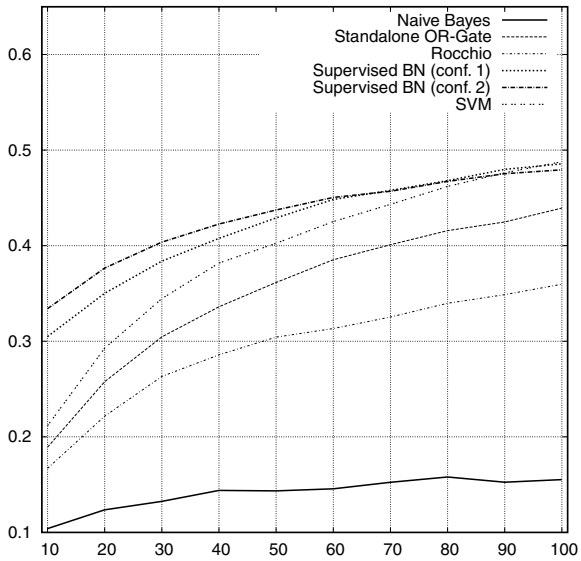

Fig. 7. Macroaveraged breakeven point computed for incremental percentage of training data.

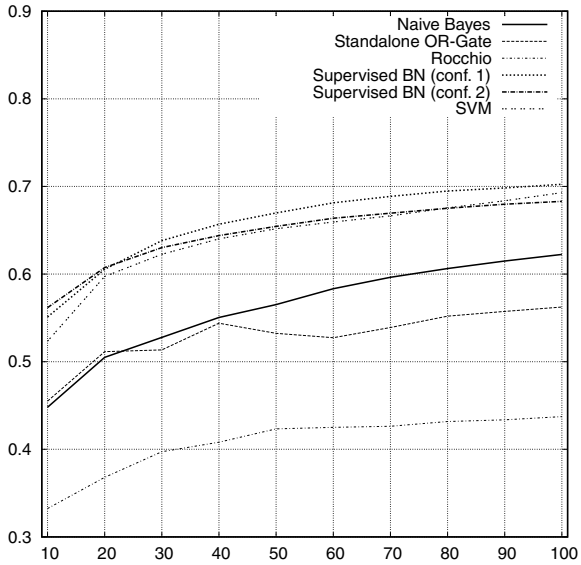

Fig. 8. Average precision at 11 standard recall points computed for incremental percentage of training data.

of the training data, in order to study if our Bayesian network model keeps its advantage with the classical models, and if it needs less data to achieve a good performance. We have selected, for comparison, two of the best performing parameter configurations, "SBN 0.5, 0.1" and "SBN 0.8, 0.1" which will be referred in the experiments as configuration 1 and 2, respectively.

For each measure (micro and macro averaged BEP, average precision and micro and macro averaged F_1 at five), we have obtained a graph, with the values of the measure at those training set percentages. In all cases, the results are also averaged over the five test partitions. The results are shown in Figs. 6–10.

The results speak for themselves: the Bayesian network model shows a great difference with two of the classical supervised approaches (Rocchio and naive Bayes) and with the OR gate model, in all the cases; in particular, when few training information is available, our model also outperforms SVM in most of the cases. Our model also tends to stabilize before and to obtain results close to the maximum in an early stage of the curve.

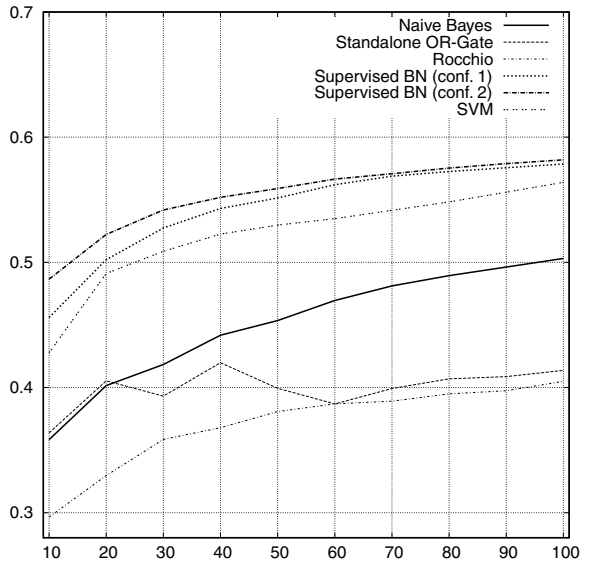

Fig. 9. Micro F_1 at five computed for incremental percentage of training data.

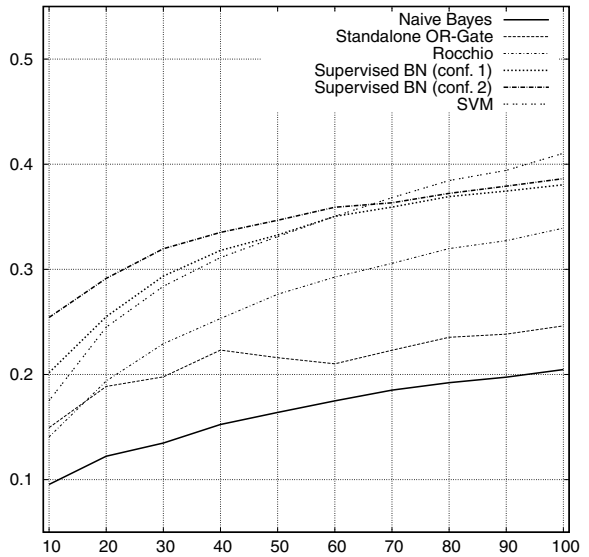

Fig. 10. Macro F_1 at five computed for incremental percentage of training data.

5. 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 resolutions from the Parliament of Andalucía and the Eurovoc thesaurus are encouraging: the model without training clearly outperforms the two simple benchmark methods considered; by integrating the initial model within a more general scheme where training data, in the form of preclassified documents, may also be used, we have also outperformed standard text classification algorithms, as Rocchio and Naive Bayes, obtaining results comparable to those of support vector machines.

For future research, we are planning to improve the initial model in two 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.

This initial model could also be combined with other supervised text classifiers different from the OR gate classifier. Perhaps the relative weights of the lexical information in the thesaurus (descriptors and non-descriptors) should depend on the

amount of training data (the more training data, the less influence of the lexical information in the thesaurus). More generally, instead of fixing manually all the parameters, i.e. the weights of the different canonical (additive and OR) probability models being used, it would be interesting to try to estimate or learn these parameters from the training data.

The Bayesian network model proposed in this paper is focused on classification using descriptors of a thesaurus. However, it could also be used in other classification problems where the different classes have associated some kind of descriptive text (which would play the role of descriptors), for example the problem of classifying documents into hierarchical web directories. Moreover, the model could also be used with a minor modification in hierarchical text classification problems, provided that the documents can be associated with internal categories (and not only with the leaves categories): by removing the virtual equivalence nodes (as well as descriptor nodes). We plan to test our model in these kinds of problems, as well as with other thesauri larger than Eurovoc, as Agrovoc or MeSH.

Acknowledgements

This work has been jointly supported by the Spanish 'Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía', 'Ministerio de Educación y Ciencia' and the research programme Consolider Ingenio 2010, under Projects TIC-276, TIN2005-02516 and CSD2007-00018, respectively.

References

[1] G. Adami, P. Avesani, D. Sona, Clustering documents in a web directory, in: Proceedings of Fifth ACM Int. Workshop on Web Information and Data Management, 2003, pp. 66–73.
[2] G. Adami, P. Avesani, D. Sona, Clustering documents into a web directory for bootstrapping a supervised classification, Data Knowledge Engineering 54 (2006) 301–325.
[3] C. Apte, F. Damerau, S.M. Weiss, Automated learning of decision rules for text categorization, ACM Transactions on Information Systems 12(3) (199) 233–251.
[4] S. Chakrabarti, B. Dom, R. Agrawal, P. Raghavan, Using taxonomy, discriminants, and signatures for navigating in text databases, in: Proceedings of the 23rd International Conference on Very Large Data Bases, 1997, pp. 446–455.
[5] L.M. de Campos, J. M Fernández-Luna, J.F. Huete, The BNR model: foundations and performance of a Bayesian network-based retrieval model, International Journal of Approximate Reasoning 34 (2003) 265–285.
[6] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, A.E. Romero, Automatic indexing from a thesaurus using Bayesian networks: application to the classification of parliamentary initiatives, Lecture Notes in Artificial Intelligence 4724 (2007) 865–877.
[7] S. Dumais, H. Chen, Hierarchical classification of web document, in: Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval, 2000, pp. 256–263.
[8] K. Golub, Automated subject classification of textual web documents, Journal of Documentation 62 (3) (2006) 350–371.
[9] T. Joachims, A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 143–151.
[10] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the European Conference on Machine Learning, 1998, pp. 200–209.
[11] T. Joachims, SVM Light Support Vector Machine, 2002, <http://svmlight.joachims.org>.
[12] D. Koller, M. Sahami, Hierarchically classifying documents using very few words, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 170–178.
[13] R.R. Larson, Experiments in automatic library of congress classification, Journal of the American Society for Information Science 43 (2) (1992) 130–148.
[14] B. Lauser, A. Hotho, Automatic multi-label subject indexing in a multilingual environment, Lecture Notes in Computer Science 2769 (2003) 140–151.
[15] D. Lewis, W. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval, 1994, pp. 3–12.
[16] A. McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, in: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, 1998.
[17] O. Medelyan, I. Witten, Thesaurus based automatic keyphrase indexing, in: Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries, 2006, pp. 296–297.
[18] R. Moskovitch, S. Cohen-Kashi, U. Dror, I. Levy, Multiple hierarchical classification of free-text clinical guidelines, Artificial Intelligence in Medicine 37 (3) (2006) 177–190.
[19] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan and Kaufmann, San Mateo, 1988.
[20] M. Ruiz, P. Srinivasan, Hierarchical text categorization using neural networks, Information Retrieval 5 (1) (2002) 87–118.
[21] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (2002) 1–47.
[22] Y. Yang, An evaluation of statistical approaches to MEDLINE indexing, in: Proceedings of the AMIA Annual Fall Symposium, 1996, pp. 358–362.
[23] Y. Yang, An evaluation of statistical approaches to text categorization, Information Retrieval 1 (1999) 69–90.