Available online at www.sciencedirect.com
SCIENCE DIRECT®
Data & Knowledge Engineering 54 (2005) 301–325
DATA & KNOWLEDGE ENGINEERING
www.elsevier.com/locate/datak

Clustering documents into a web directory for bootstrapping a supervised classification

Giordano Adami, Paolo Avesani, Diego Sona *

ITC-IRST, Automatic Reasoning Systems Division, Via Sommarive 18, 38050 Povo, Trento, Italy

Received 4 November 2004; received in revised form 4 November 2004; accepted 18 November 2004
Available online 8 December 2004

Abstract

The management of hierarchically organized data is starting to play a key role in the knowledge management community due to the proliferation of topic hierarchies for text documents. The creation and maintenance of such organized repositories of information requires a great deal of human intervention.

The machine learning community has partially addressed this problem by developing hierarchical supervised classifiers that help people categorize new resources within given hierarchies. The worst problem of hierarchical supervised classifiers, however, is their high demand in terms of labeled examples. The number of examples required is related to the number of topics in the taxonomy. *Bootstrapping* a huge hierarchy with a proper set of labeled examples is therefore a critical issue.

This paper proposes some solutions for the *bootstrapping* problem, that implicitly or explicitly use taxonomy definition: a *baseline* approach that classifies documents according to the class terms, and two clustering approaches, whose training is constrained by the a priori knowledge encoded in the taxonomy structure, which consists of both terminological and relational aspects. In particular, we propose the *TaxSOM* model, that clusters a set of documents in a predefined hierarchy of classes, directly exploiting the knowledge of both their topological organization and their lexical description. Experimental evaluation was performed on a set of taxonomies taken from the Google™ and LookSmart™ web directories, obtaining good results.

* Corresponding author.
E-mail addresses: gioadami@itc.it (G. Adami), avesani@itc.it (P. Avesani), sona@itc.it (D. Sona).
URL: http://sra.itc.it

0169-023X/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2004.11.003

Keywords: Web directories; TaxSOM; Constrained clustering; *K*-means; Unsupervised learning; Relational machine learning; Taxonomy bootstrapping; Text categorization; Knowledge management; Digital libraries

1. Introduction

Recent trends in knowledge management highlight the interest on the organization of documents or other sources of knowledge into hierarchies of concepts [6]. Web directories represent a widespread scenario where the most relevant web pages are classified with respect to a predefined set of categories organized into a hierarchy. Google™ [16], Yahoo!™ [30] and LookSmart™ [22] are well-known examples of such hierarchical organization of knowledge. This categorization approach is strategic within company Intranets, too, because knowledge management platforms very often support the hierarchical organization of information. Actually, taxonomic structures are considered a shallow representation of knowledge. The interest and the relevance of document organization into taxonomies is well represented by the *dmoz.org* initiative, an open source initiative raised to promote a comprehensive web directory (see the Open Directory Project [11]).

A taxonomy is mainly defined by two components: a hierarchy of categories and a collection of documents. Each node detects a category, and the categories are described both by linguistic labels that denote the "meaning" of the nodes, and by the relationships with other categories. Documents are classified under one or more categories according to a single or multiple organization strategy. Most of the time, hierarchical indexes are treated as discrimination trees, where the intermediate nodes do not necessarily refer to a specific category. In this kind of hierarchical structure, documents are annotated with respect to the leaves only. Taxonomies and web directories differ from such hierarchical indexes, because even interior nodes in the hierarchies refer to a specific concept. Documents are also annotated to interior nodes.

Document annotation is a typical task in the management of web directories. Given a predefined taxonomy, the goal is to identify the category related to the content of an unclassified document. In this perspective the categorization task can be conceived as a problem of finding the right location in the hierarchy. Many supervised document classifiers that enable the automation of this task have been designed (see for example [7–9,12,13,18,20,26,27,29]). However, most of them have a common restrictive precondition: for each category a training set made of a significant amount of labeled documents is required. This issue is known in literature as the *bootstrapping* problem [23].

Bootstrapping a hierarchical structure of categories with a correct set of labeled examples is a critical point in the deployment of automated classifiers. Actually, the number of labeled examples required to train a supervised learning algorithm is related to the size of the taxonomy. For example, the most popular web directories, like Google, Yahoo!, and LookSmart, have large hierarchical structures with many thousands of nodes, i.e., categories. It is worthwhile to remember that hierarchies nodes tend to grow exponentially.

Although in real world scenarios the creation of structured indexes is an evolving process (i.e., document classification and structure definition are interleaved steps), this paper is focused on the early stage of the process. In this stage, a *bootstrapping* model plays a key role in supporting the preliminary annotation of a taxonomy. In the beginning, the taxonomy is empty, i.e., there are no

categorized documents. Hence, the tool used to automatically annotate the documents has no annotation examples. A first categorization hypothesis has to be formulated only using the taxonomy definition.

It follows that the output of the *bootstrapping* model is a labeling hypothesis for a given set of candidate documents. The ultimate goal of this model is to meet the preconditions required to successfully train a supervised classifier. There is no need for a highly accurate unsupervised classification. The real challenge is to semi-automatically determine the proper amount of labeled examples for each category, thereby reducing the user effort required to classify the documents.

Analysing the *bootstrapping* problem from scratch, the perspective is twofold. The task can be conceived as a classification problem, since an hypothesis of classification for a set of unlabeled documents must be given on the basis of the node's labels. At the same time, it can be conceived of as a constrained clustering task, where the number, the type, and the relationships of clusters are defined in advance. In the following, the paper will propose how to reconcile these two conflicting perspectives to effectively support the *bootstrapping* process.

We conceive *bootstrapping* as a task within a more complex process where machine and human effort are interleaved. The basic idea is to support and alleviate the manual labeling of a set of unlabeled examples, providing the user with an automatically determined preliminary hypothesis of classification. The idea is to exploit the linguistic and the relational information encoded within a taxonomy through an unsupervised learning model. The paper illustrates how *Self-Organizing Maps* (SOMs) [19] can be revised to influence the learning process with the knowledge encoded within a taxonomy.

In Section 2 a model of the *bootstrapping* process is introduced. This model covers all the steps underlying the deployment of a fully automated hierarchical document classifier. Section 3 illustrates the unsupervised model referred to as *TaxSOM*, used to provide a preliminary classification. Section 4 describes the datasets used to test the models, their preprocessing and encoding, and the criteria used to evaluate the models. Results of the model evaluation and the respective discussion are reported in Sections 5 and 6 respectively. Finally, Section 7 presents a discussion on the differences between the proposed approach and previous solutions proposed in the literature.

2. The bootstrapping process

The term *bootstrapping* refers to the sequence of steps that, starting from an empty taxonomy of concepts, enables the delivery of an automated document classifier. Detailing the process permits us to understand how human and machine roles can be combined to decrease manual effort and to increase the quality of the final classification result. Fig. 1 sketches a *bootstrapping* process, detailing the different steps and the intermediate results. The diagram shows the tasks assigned to user and machine.

Step 1: *Taxonomy editing.* The first step of the process consists in the definition of a taxonomy of concepts, where the categories are encoded through linguistic labels and the hierarchical relationships are encoded into a *tree-structured directed graph*. For an example, see the left-hand side graph of Fig. 2. After this stage, the system has an empty taxonomy that needs to be "populated" with a set of candidate documents.

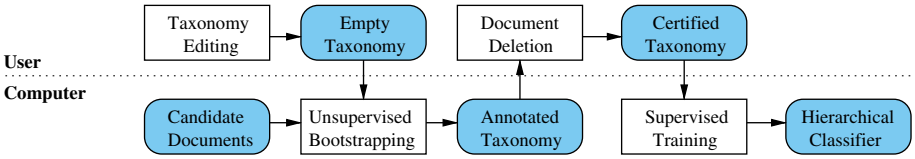

Fig. 1. A model of the *bootstrapping* process. The schema illustrates a mixed-initiative strategy combining machine and human effort.

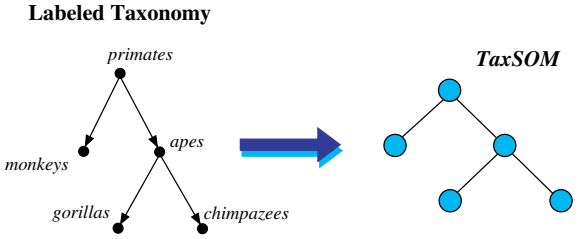

Fig. 2. *TaxSOM* is a graph of computational units connected according to the topology of the given taxonomy. The above taxonomy is a snapshot of a small part of the Google web directory.

Step 2: *Unsupervised bootstrapping.* The second step concerns the elaboration of a preliminary labeling hypothesis for the candidate documents. In this phase, the goal of the task is to locate (classify) the documents in a node of the taxonomy. The result of this *bootstrapping* step is an annotated taxonomy, i.e., a hierarchy of classes where all the candidate documents have been classified in a corresponding node (category) of the taxonomy.

Step 3: *Document deletion.* In this phase, the expert (editor of the taxonomy) manually checks the hypothesis formulated by the machine. The expert has to deal with two simple alternatives: confirm the classification hypothesis for a given document or discard it. At the end of this step, a certified annotated taxonomy is obtained, with a set of labeled documents for each category.

Step 4: *Supervised training.* The last step concerns the supervised training of a hierarchical classifier, which can use both the taxonomic information and the set of labeled examples.

The core problem of the *bootstrapping* process consists on the unsupervised step that can be formally defined as a function taking an empty taxonomy and a set of documents as input, and returning an association between documents and nodes of the taxonomy.

The working hypothesis underlying the proposed approach is that, once presented with a document, the taxonomy editor would prefer to deal with a Boolean decision (confirm/discard) on the given labeling hypothesis rather than examining all possible labeling alternatives. Notice that the sketched model does not specify the deletion policy. Actually, many different policies are allowed. For example, it is conceivable to ask the user for a selection between various alternatives when the categorization hypothesis is wrong. Anyway, considering complex strategies requires to model the cognitive effort of the taxonomy editor, and this is out of the scope of the paper.

Therefore, we are not interested in the process from the user point of view. Rather, we evaluate the performance from the machine perspective. The objective of the *bootstrapping* process is two-

fold: produce a good set of correct hypotheses, i.e., a significant number of correctly classified examples, and achieve homogeneous coverage of all the categories. Hence, the ultimate goal of the unsupervised model is the production of the proper amount of labeled examples for each class: enough to train a supervised classifier without the need for a highly accurate hypothesis of classification. Of course, the overall performance of the *bootstrapping* process is a trade-off between the amount of correctly labeled examples and the effort required to validate (by hand) the classification hypothesis. This paper does not consider the potential for reducing the classification hypothesis adopting a rejection criterion in case of categorization ambiguity.

The *bootstrapping* process aims to be independent of the specific supervised classifier, and it is not a goal of this paper to outperform existing supervised models. The goal is to assess whether, given a supervised classifier such as a simple nearest neighbor rule, the supervised training phase is more effective using a *bootstrapping* technique based on *TaxSOM* model rather than a *baseline* solution. For the sake of exposition, any complex variation of the process, such as the introduction of feedbacks repeating all steps as many times as needed, will not be considered.

The model illustrated above does not make any assumption on the source of the candidate documents. Two alternative scenarios can be conceived making different assumptions:

- an *open-world* scenario traditionally related to the web;
- a *closed-world* scenario more related to the Intranets domain.

The two scenarios differ on a basic assumption. In the former case, only a portion of all unlabeled documents are supposed to be related to the categories of the given taxonomy. In the latter case, a strong hypothesis holds: all candidate documents are related to at least one category of the given taxonomy. Clearly, the two different cases strongly influence the process. In the first scenario, the problem is twofold. On the one hand, a ''macro categorization'' to assess the relevance of a document with respect to the domain of the taxonomy is required. On the other hand, once a document is found to be related to the taxonomy domain, a ''micro categorization'' to solve the class ambiguity is required.

The *closed-world* scenario clearly does not require the ''macro categorization'', since all documents are known to belong to the taxonomy domain. A plausible example of this scenario is represented by a company promoting a revision of an internal structured organization of a given collection of documents. A new taxonomy is arranged and, afterward, all documents need to be indexed accordingly.

For simplicity, this second scenario will be examined. The assumption is that all the candidate documents are related to at least one category of the given taxonomy. Therefore, the focus will be on solving the ambiguity among different categorization alternatives rather than filtering documents not related with the topics of the taxonomy.

3. Exploitation of relational knowledge

The challenge of the *bootstrapping* task is the exploitation of the a priori knowledge encoded in a taxonomy, that is, the linguistic labels describing the meaning of categories, and the topological

structure referring to the relationships among categories. More formally, a taxonomy can be defined as follows:

Definition 1 (*Taxonomy*). A taxonomy $\mathscr{T} = (N, E, L_N)$ is a labeled directed graph, where N and E are finite sets of nodes and directed edges connecting nodes respectively, while L_N are node labels.

The nodes of the graph correspond to the categories (sometimes referred to as classes or concepts as well), and the edges describe the relationships between the categories. A further assumption is that labels (L_N) are taken from the lexicon of natural language. In particular:

Definition 2 (*Features set*). Given a taxonomy $\mathscr{T}$ and a set of documents $\mathscr{D}$, the feature set $\mathscr{F}^{\mathscr{T},\mathscr{D}}$ is the set of all lexicon words appearing as either node labels in $\mathscr{T}$, keywords in the documents of $\mathscr{D}$, or both.

and

Definition 3 (*Labels and keywords sets*). Given a taxonomy $\mathscr{T}$ and a set of documents $\mathscr{D}$, the label set L_N is the set of lexicon words appearing as node labels in $\mathscr{T}$, and the keywords set L_D is the set of lexicon words appearing in the documents of $\mathscr{D}$ but not as node labels of $\mathscr{T}$. Hence, $L_{\mathscr{D}} \cup L_N = \mathscr{F}^{\mathscr{T},\mathscr{D}}$ and $L_D \cap L_N = \emptyset$.

In order to both simplify the computational complexity and increase the models generalization during learning, a "reduced" vocabulary needs to be derived, performing a feature selection on the above two sets L_N and L_D. This vocabulary is then used to encode the documents of dataset $\mathscr{D}$. A first important constraint on the vocabulary creation is that the resulting vocabulary must contain all the keywords used to label the nodes. Another constraint is that the feature selection of document keywords can only be performed with unsupervised techniques, since all documents are unlabeled.

Definition 4 (*Vocabulary for TaxSOM*). A vocabulary $\mathscr{V}^{\mathscr{T},\mathscr{D}}$ for a given taxonomy $\mathscr{T}$ and a set of documents $\mathscr{D}$ is a subset of $\mathscr{F}^{\mathscr{T},\mathscr{D}}$ that contains all keywords in L_N and a subset of keywords in the documents. Hence, $\mathscr{V}^{\mathscr{T},\mathscr{D}} = L_N \cup L'_{\mathscr{D}}$ where $L'_{\mathscr{D}} \subseteq L_{\mathscr{D}}$ is determined with a feature selection algorithm.

The main aim of this work is to find a model that—given a taxonomy $\mathscr{T}$, a document set $\mathscr{D}$, and a vocabulary $\mathscr{V}^{\mathscr{T},\mathscr{D}}$—returns a preliminary classification of documents in $\mathscr{D}$ according to the a priori knowledge on $\mathscr{T}$. Therefore, we propose a new clustering model referred to as a *Taxonomic Self-Organizing Map* (*TaxSOM*). The model and its training algorithm are partially derived from the *Self-Organizing Maps* (SOMs) [19]. This new model organizes data in $\mathscr{D}$ according to a given taxonomy $\mathscr{T}$, using selected information encoded in the vocabulary $\mathscr{V}^{\mathscr{T},\mathscr{D}}$. Notice that letting the vocabulary be part of the model input makes the classification process and the feature-selection process orthogonal to each other.

In contrast to the original SOM model, the topology describing the class relationships and the set of labels describing the meaning of categories are part of the model input, and they are used during learning as a bias on the way to exploit the contextual information. The basic idea is that a taxonomy consists of a structured set of classes, related to each other according to a fixed topology. This topology influences the training algorithm of the *TaxSOM* model. Moreover, the labels

are also used to constrain the adaptation of the codebooks. [1] Therefore, forcing these two features into the learning algorithm can be considered a way to perform a *constrained clustering* of documents. In the following, a brief review of some basic notions of SOM is given, following which the *TaxSOM* model is presented.

3.1. Review of self organizing maps

A SOM consists of k computational units located on a regular low-dimensional grid $\mathscr{A}$, usually planar with rectangular or hexagonal connection schemes. Each unit is described by both a position index in the lattice, and a codebook vector $\vec{w}_i = [w_1, \ldots, w_m]$, which is a cluster centroid in the input space. SOMs are trained by alternating between a *competitive* and a *cooperative* phase for each input pattern. During the *competitive* stage, the codebook most similar to the input vector $\vec{x}$ is chosen as the *winner* unit (like in a standard prototype-based minimum error classifier, where documents are related to the cluster having the nearest prototype):

$$i^* = \arg\max_i \{\mathrm{sim}(\vec{w}_i, \vec{x})\}, \tag{1}$$

where $\mathrm{sim}(\vec{w}_i, \vec{x})$ computes the similarity measure between the two objects $\vec{w}_i$ and $\vec{x}$. In our experiments, we adopted the cosine similarity metric:

$$\mathrm{sim}(\vec{w}_i, \vec{x}) = \frac{\vec{w}_i^t \cdot \vec{x}}{\|\vec{w}_i\| \cdot \|\vec{x}\|}. \tag{2}$$

In the *cooperative* stage, all codebooks are moved closer to the input vector, with a learning rate (non-linearly) proportional to the inverse of their topological distance from the winner unit:

$$\vec{w}_i(t+1) = \vec{w}_i(t) + \eta(t) h_{i,i^*}(t)[\vec{x} - \vec{w}_i(t)], \tag{3}$$

where $\eta(t)$ is the learning rate, and $h_{i,i^*}(t)$ is a neighborhood function monotonically decreasing for increasing topological distance between unit i and the winner unit i^*. Usually, the neighborhood function $h_{i,i^*}(t)$ is a Gaussian function with decreasing variance:

$$h_{i,i^*}(t) = \exp\left(-\frac{\mathrm{dist}(i, i^*)^2}{2\sigma(t)^2}\right), \tag{4}$$

where $\sigma(t)$ is the function range (width of the neighborhood) decreasing in time, and $\mathrm{dist}(i,i^*)$ is the topological distance between the two units i and i^* in the discrete lattice $\mathscr{A}$. Usually in the SOM model such distance is computed as follows:

$$\mathrm{dist}(i, i^*) = \|\vec{r}_i - \vec{r}_{i^*}\|, \tag{5}$$

where $\vec{r}_i$ and $\vec{r}_{i^*}$ are the coordinates of the two units i and i^* respectively.

[1] Latter in this article, the terms codebook, prototype, and reference vector are used intermittently with the same meaning.

3.2. The TaxSOM model

The main property of SOMs is that similarity relationships between patterns in the input space are mapped into similarity relationships between codebooks. Specifically, similar documents are mapped to the same unit or to near units in the lattice.

The idea of *TaxSOM* is to exploit this property, modeling the connections between the computational units according to the topology of the input taxonomy. Specifically, given a taxonomy $\mathscr{T}$, a *TaxSOM* is a collection of computational units connected so as to form a graph having the shape isomorphic to the topology of $\mathscr{T}$. For each node in $\mathscr{T}$, a computational unit is created in *TaxSOM*. For any directed edge connecting two nodes in $\mathscr{T}$, an undirected edge connecting the corresponding units in *TaxSOM* is created (see example in Fig. 2).

The conjecture is that once a *TaxSOM* has been trained by iterating on Eq. (3), the final configuration of the codebooks describes a clustered organization of documents—that tailors the desired relationships between concepts. Nevertheless, the model only exploits the topological organization of classes and does not exploit the lexical information. A simple way to also handle lexical information is derived from a standard artifice used to speed up the clustering processes: The trick is to start the algorithms with the codebooks properly initialized [19].

A good starting point for codebooks is based on the exploitation of the a priori lexical knowledge, i.e., the labels in L_N "describing" the concepts in the taxonomy $\mathscr{T}$. More specifically, for any node in *TaxSOM* an initial codebook (a reference vector also referred to as "seed") is built through the encoding of its labels, i.e., all elements of a given codebook are set to zero except those elements that correspond to the node labels. In particular, because in the current task documents are represented by a *set-of-words*, [2] the elements of codebooks corresponding to the node labels are set to 1. Notice that both documents and codebooks are represented with vectors of length equal to the vocabulary dimension.

We observed that the described initialization criterion let *TaxSOM* quickly converge to a solution though one of poor quality in the sense that the node coverage [3] and the classification hypothesis are of low quality. Actually, we noticed that the algorithm starts the clustering process with a "good" classification accuracy, but during training the organization of documents changes according to criteria that are different from classification purposes. This causes poor results. The main problem is the lack of constraints during training.

To minimize inter-category variances while trying to preserve good classification accuracy, the algorithm was modified to constantly preserve the lexical information in the codebooks. Specifically, the labels are forced in the codebooks throughout the entire training process. This occurs by encoding the labels in the codebooks after each *competitive* step, setting to 1 all elements of the codebooks corresponding to the concept labels. In this way, *TaxSOM* training is biased by the knowledge of both the concepts relationships (topology), and the concepts descriptions (labels).

In the current implementation of *TaxSOM*, in order both to reduce the computational cost of the simulations while addressing large data sets, and to smooth the convergence to a solution with low variance, we designed a variation of the batch training algorithm originally designed for

[2] *Set-of-words* corresponds to a binary representation that identifies the presence or the absence of the corresponding keywords in a given vocabulary.

[3] Node coverage is a measure of how well nodes are populated with correct examples.

SOMs [19]. The batch algorithm requires a smaller number of iterations with respect to the one described above. This is because the *competitive* phase is performed on the whole data set before computing the *cooperative* step, and the *cooperative* phase is then performed with two computations per iteration.

The computation of any iteration of the proposed batch algorithm can be divided into three main steps. In the first step, for any node i of a given taxonomy $\mathscr{T}$, the training algorithm of *TaxSOM* computes a centroid $\vec{c}_i(t)$. The centroid is determined averaging the documents that in the previous iteration were classified in the corresponding node:

$$\vec{c}_i(t+1) = \frac{1}{n_i(t)} \sum_{\vec{x} \in \mathscr{D}_i(t)} \vec{x}, \tag{6}$$

where $\mathscr{D}_i(t)$ is the *Voronoi set* of class i at time t (i.e., the set of documents classified in class i), and $n_i(t)$ is the number of documents in $\mathscr{D}_i(t)$.

In the second step, a smoothing procedure is carried out on the centroids, obtaining an "unconstrained codebook":

$$\vec{w}_i^u(t+1) = \frac{\sum_j n_j(t) \cdot h_{i,j}(t) \cdot \vec{c}_j(t+1)}{\sum_j n_j(t) \cdot h_{i,j}(t)}, \tag{7}$$

where $h_{i,j}(t)$ is the neighborhood function described by Eq. (4), whose distance measure dist(i,j) is the length of shortest path between the two nodes i and j in $\mathscr{T}$. Because the structures processed are trees, the distance corresponds to the sum of the node depths with respect to their nearest common ancestor in the hierarchy. Notice however that the model can also be used for graphs with cycles.

The constraining phase is then performed in a third step, where the "constrained codebooks" are computed by encoding the node labels into the corresponding "unconstrained codebooks". This operation is carried out with the function $f : \mathbb{R}^n \times \mathbb{L} \to \mathbb{R}^n$ that for any node i in $\mathscr{T}$ takes as input an "unconstrained codebook" $\vec{w}_i^u(t)$ and a set of keywords L_{N_i} describing the corresponding concept, and returns the final codebook $\vec{w}_i(t)$ constrained as follows:

$$\vec{w}_i(t) = f(\vec{w}_i^u(t), L_{N_i}) \text{ s.t. } \forall j \quad w_{i,j}(t) = \begin{cases} 1 & \text{if } \mathscr{V}_j^{\mathscr{T},\mathscr{D}} \in L_{N_i} \\ w_{i,j}^u(t) & \text{otherwise} \end{cases}, \tag{8}$$

where $\vec{w}_i(t)$ and $\vec{w}_i^u(t)$ are respectively the constrained and unconstrained codebook vectors of node i at time t, L_{N_i} is the corresponding set of labels, $\mathscr{V}_j^{\mathscr{T},\mathscr{D}}$ is the jth keyword in the vocabulary, and $w_{i,j}(t)$ and $w_{i,j}^u(t)$ are the jth elements of vector $\vec{w}_i(t)$ and $\vec{w}_i^u(t)$ respectively. Notice that the initial point used to start *TaxSOM* learning (time $t = 0$) is determined using Eq. (8) with $\vec{w}_{i,j}^u(0) = 0 \ \forall i, j$.

The above learning equations constrain the codebooks with the *a priori* knowledge localized on the nodes, and the contextual information [4] is gathered by propagating the "documents content" from the nodes in the neighborhood (Eq. (7)). Notice that the node labels are not propagated to

[4] The contextual information for a given node is the knowledge that can be collected from the nodes in its neighborhood.

the neighbor nodes, but their propagation could be useful as well. Hence, in order to also propagate this type of information Eq. (7) needs to be changed as follows:

$$\vec{w}_i^u(t+1) = \frac{\sum_j n_j(t) \cdot h_{i,j}(t) \cdot \vec{c}_j^c(t+1)}{\sum_j n_j(t) \cdot h_{i,j}(t)}, \tag{9}$$

where $\vec{c}_j^c(t+1)$ is the jth centroid constrained by the set of labels of the corresponding node (L_{N_i}):

$$\vec{c}_j^c(t+1) = f(\vec{c}_j(t+1), L_{N_j}), \tag{10}$$

where $\vec{c}_j(t+1)$ is computed using Eq. (6). Therefore, using Eqs. (8)–(10) labels are propagated to the classes in the neighborhood, and are also used to constrain the codebooks.

Using the above learning equations, the codebooks start their learning from "good" initial points in the input space determined by the set of labels of the corresponding nodes. Then, during training, the prototype vectors learn a different position in the input space that better represents a good clustering [5] of the input space. Interestingly, while the codebooks are able to learn an optimal solution for the clustering task, they are continuously attracted by their initial point, maintaining a linking between clusters and classes. This trade-off between constraints and clustering of documents allows the codebooks to learn a solution which is constrained to be good for the classification task.

In fact, when a clustering algorithm is used to classify, it is important to discover the correlation between the desired classes and the clusters found with the algorithm. The proposed model, on the contrary, anchors the clusters to the classes, forcing the required linking from the beginning, and maintaining these links throughout training.

4. Experimental setup

The unsupervised *bootstrapping* of a taxonomy is achieved through a function that takes the taxonomy and a set of documents as input, and returns an association between documents and nodes of the taxonomy. Since the task output should be a "significant" dataset of labeled documents, [6] the proposed model needs to be evaluated by looking at both the classification accuracy and the quality of the labeling produced. For this reason the model was evaluated with two different criteria:

- The *TaxSOM* model was evaluated against other basic models looking at the accuracy of the classification results.
- The model was evaluated looking at the quality of the whole process that combines unsupervised and supervised models. Specifically, the accuracy of a simple non-parametric supervised model was evaluated on the results of *TaxSOM*. This was to verify how the results of the unsupervised bootstrapping model influence the whole process.

The description of the evaluation methods together with a detailed description of the datasets used to evaluate the model follows in this section.

[5] In this case the term "good" is related to the concept of classification.

[6] Significant dataset means that the set of correctly classified examples generated by the model is statistically meaningful in terms of the quality of the supervised model training.

4.1. The datasets

In order to experimentally evaluate the *TaxSOM* model, a set of labeled and annotated taxonomies were needed. Specifically, the evaluation should be performed using hierarchies of classes, where each node (both interior and leaves) is labeled with linguistic keywords describing the class content, and populated with a set of labeled documents. Web directories are meaningful examples of such type of taxonomies. Hence, to evaluate the proposed model in a real-world scenario, we created a benchmark dataset made up of a set of taxonomies selecting some domains from two well-known web directories: Google [16] and LookSmart [22].

Specifically, the selected taxonomies are trees corresponding to some subdirectories of Google and LookSmart. The labels of the nodes in the taxonomies are the names of the nodes in the directories (usually a few keywords). The documents are the URL descriptions built using the web site title and the short summary given by the directory maintainers (usually a text of a few dozen words), see the example in Fig. 3.

Furthermore, within LookSmart taxonomies, each node is also equipped with a short description of the node content. These few keywords can be used to partially define the concept semantic of the corresponding class, which is much more detailed that the description provided by the node labels. In the following experiments, these natural language words have been considered as labels. In this way, evaluating the models with these two datasets (Google and LookSmart) it is possible to verify how the number of labels used to constrain the learning procedure influence the quality of the model.

Such subdirectories were chosen ranging over many different topics and dimensions (see Table 1). Taxonomies were selected looking at their depth (i.e., how far leaves are from the root, which ranges from 4 to 11), looking at the number of nodes (from tens to hundreds), and looking at the number of documents (from hundreds to thousands). The topic and dimension variability allows the evaluation of the model without biases due to the a priori knowledge, the topic vocabularies, or the dimension of taxonomies.

directory: Google: Home > Cooking
node: cooking/soups_and_stews/fish_and_seafood
labels: fish, seafood

document 1
URL: http://www.fish2go.com/rec_0120.htm
excerpt: Finnan Haddie and Watercress Soup: Made with smoked haddock, potatoes, watercress, and milk.
lemmata: haddock, make, milk, potato, smoke, soup, watercress.

document 2
URL: http://www.bettycrocker.com/default.asp
excerpt: Crunchy Snacks from Betty Crocker: Collection of sweet and savory snack recipes which pack a crunch, from healthy vegetables to s'mores.
lemmata: collection, healthy, recipe, savory, snack, sweet, vegetable.

Fig. 3. Example of two "documents" in the "Fish & Seafood" node of Google. All document lemmata and node labels are used to create the vocabulary and to encode documents. Notice that documents contain very few keywords, and only two words are used as node labels.

Table 1
Statistics of Google and LookSmart benchmarks

	Taxonomy statistics				Dataset statistics		Docs with labels (%)		
	Max depth	Nodes	Docs	Labels	Vocabulary size	Average words/doc	Local	Ancestors	Global
Google									
Archaeology	5	122	1204	201	650	10.93	44	70	94
Biology	11	1601	9213	1528	1921	8.69	60	73	100
Business	5	213	7563	233	586	9.07	62	85	99
Cooking	7	674	16,318	594	862	7.93	75	89	100
Language	8	514	4134	488	864	8.93	64	90	100
Neuro Disorders	4	210	2444	231	631	9.68	70	89	98
News Media	5	29	549	35	585	10.68	52	71	88
Shopping Health	6	259	8652	312	645	8.29	58	77	99
Technology	7	571	8295	509	779	9.14	59	81	100
LookSmart									
Archaeology	5	78	794	274	701	8.61	80	90	100
Business Soft.	8	276	6196	519	728	9.41	93	97	100
Common Lang.	4	140	1890	349	639	9.01	95	97	100
Health Issues	9	528	6244	728	924	9.29	95	98	100
Linguistics	8	319	2565	887	1114	8.14	93	96	100
Movies	4	34	682	139	825	10.30	95	98	100
Peripherals	5	198	4574	414	672	11.41	96	98	100
Recipes	8	599	8906	841	994	8.59	98	99	100
Videogames	7	417	3685	768	919	10.90	98	99	100
Zoology	9	1007	10,927	1549	1645	9.64	95	99	100

The first four columns describe the dataset dimension, the other two columns are devoted to the evaluation of the documents encoding, the remaining columns describe the task complexity, i.e., how ambiguous the classification might be when just driven by labels.

The selection of the taxonomies was constrained by some homogeneity criteria. In particular, all documents having less that five keywords after preprocessing and feature selection were removed from the dataset. All empty nodes (nodes without documents) or unlabeled nodes (nodes without any label having lexical meaning) and their sub-trees were removed from the taxonomies. Finally, notice that some taxonomies with similar topics were selected from the two web directories as well, e.g., "*archaeology*", "*language*" and "*linguistics*", "*biology*" and "*zoology*", etc. This allows a further comparison of different organizations of data within similar topics. For a detailed description of the benchmark dataset used for the experiments refer to [4].

4.1.1. Feature selection and document encoding

Document content and node labels were cleaned. In particular, all stop-words (articles, conjunctions, and prepositions) were removed, and the remaining words were stemmed (transformed into lemmata). See labels and lemmata in Fig. 3 for an example of features preprocessing.

The space of the encoding vector, i.e., the vocabulary, was separately determined for each taxonomy as previously defined (see Definition 4). Since one of the constraints on the vocabulary

determination is that the feature selection techniques can only be unsupervised, we adopted a feature selection process based on the notion of Shannon Entropy. [7] In particular, the adopted feature selection process reduces the set L_D to a subset L'_D keeping the words with the highest entropy defined as follows:

$$\mathscr{H}(X) = -\sum_{x\in\{0,1\}} p(x)\log_2 p(x) \tag{11}$$

where X is the considered feature and x is the possible value assumed by the variable X in the given dataset. The creation of a vocabulary with the features having high entropy guarantees that the keywords with very low or very high frequencies are not used, since they are neither representative nor discriminative. Clearly, probabilities used in Eq. (11) were estimated using the frequencies of the keywords in dataset.

The proposed models were tested using different numbers of features. From the experiments, it appeared that good accuracy is possible by adopting a dimension for L'_D of approximately 500 keywords. The reference vocabularies $\mathscr{V}^{\mathscr{T},\mathscr{D}}$ were then determined according to Definition 4.

Finally, due to the usually negligible number of word repetition in document descriptions (see the sets of lemmata in the examples of Fig. 3), documents were encoded as *set-of-words*, i.e., binary vectors $\vec{x}$ with dimension equal to the cardinality of $\mathscr{V}^{\mathscr{T},\mathscr{D}}$, where each element x_i of an encoded document indicates whether the corresponding keyword is present in the document or not (1 or 0).

Table 1 illustrates some statistics of the selected taxonomies after preprocessing. The first section (Taxonomy statistics) provides information on the size of both the taxonomies and the corpora of documents and the set of node labels. The second section (Dataset statistics) illustrates the vocabulary dimension, and the average number of keywords per document. The third section (% of docs with labels) gives a flavor of the problem complexity. The column "Local" is the percentage of documents in which at least one label of the corresponding class appears. The column "Ancestors" gives the same measure considering all labels in the path from the root node to the current node. Finally, the column "Global" is the percentage of documents having at least one of the labels in L_N.

Notice that the percentages of labels in documents are rather high (mostly near 100%), while the "ancestor" and "local" label occurrences are significantly lower. This causes high ambiguity when trying to classify documents just using the labels occurrences. This in turn produces high rejection rates.

4.2. Evaluation method

As previously outlined, *TaxSOM* is a module that should be embedded in a more complex process made up of three main components:

- a *bootstrapping* module that produces a preliminary hypothesis of classification for a given unlabeled dataset;

[7] Shannon entropy is a standard information theoretic approach that can be used to measure the amount of information provided by the presence of a word in the dataset.

- a cleanup step carried out by a human expert that, in the simplest approach, removes all badly classified documents;
- a classification module performed with a supervised model that classifies new, incoming documents using a "certified" labeled dataset.

Therefore, the proposed *bootstrapping* model needs to be evaluated considering two aspects. On the one hand, the classification accuracy of the model is of primary importance, since high accuracy of the model implies minimal human effort in checking the correctness of the document classification. On the other hand, the quality of the cleaned-up dataset needs to be considered. In fact, it influences the accuracy of the supervised model. Even if the model itself has low accuracy, it could generate a good training set for supervised models.

To our knowledge there are no models devised to solve the proposed task. For this reason, in order to evaluate the solution provided, we devised a couple of approaches referred to as *baseline* and *constrained K-means*. Comparing *TaxSOM* with these models allows the analysis of the influence on the model of all the elements of the available a priori knowledge.

The model accuracy was tested on each benchmark taxonomy performing an hypothesis of classification for all documents, and the results were then compared with the original labeling of the documents.

The influence of the proposed models on the third step of the *bootstrapping* process (the supervised classification) was assessed adopting a 10-fold cross validation technique. Specifically, for any taxonomy, 90% of the documents were used to bootstrap the taxonomy. Then, the resulting annotated taxonomy was automatically "cleaned" by removing all wrongly classified documents. [8] Finally, the remaining 10% of the documents were classified using a very simple supervised model (1-*NN*), with the inductive base made up of the cleaned dataset. The results were then averaged over all 10 folds. Remember that a good performance with the supervised classifier was not the aim of this work. The main intent was to evaluate how the bootstrapping models influence a supervised model.

Notice that in the experiments single-class labeling is performed.

4.3. Evaluation measures

The proposed models were assessed looking at both the accuracy and the quality of classification. The evaluation was performed considering the proposed models both as stand-alone algorithms and as part of the bootstrapping process. Therefore, different criteria for the assessment of the model were used, according to the type of evaluation.

Specifically, the stand-alone comparison of the proposed models (i.e., *baseline*, *K-means*, and *TaxSOM*) was performed using the standard information retrieval measure $F1$ [5], which combines *precision* and *recall* of a model on a given dataset:

[8] The automatic cleaning of the annotated taxonomy should be consonant with the expert behavior. Therefore, a simulation of a very simple accept/reject decision that could be made by an expert was performed. Specifically, the system eliminated all those documents from the bootstrapped taxonomy that, according to the original labeling, were incorrectly classified.

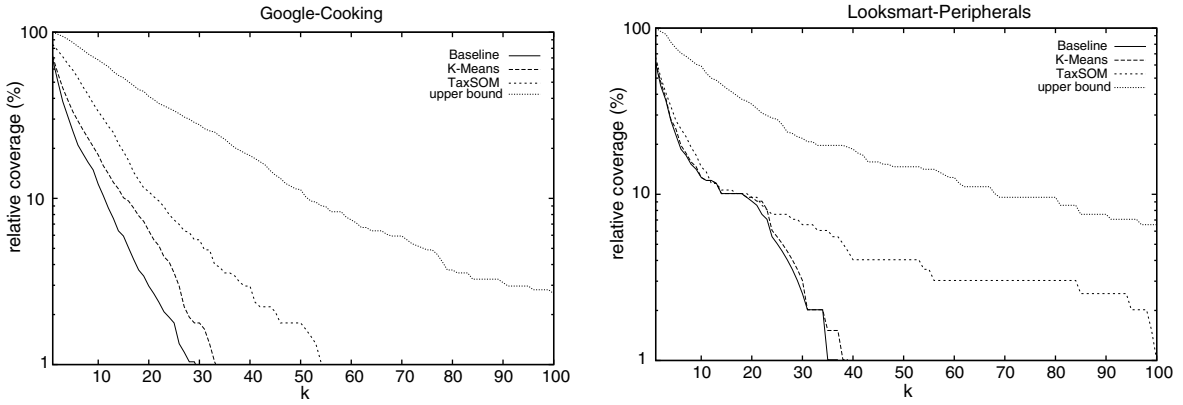

Fig. 4. Examples of *k-relative coverages* for all k for two taxonomies. The curves are plotted for the dataset resulting from the three bootstrapping model and for the original dataset (upper bound).

$$F1 = \frac{2 \cdot P \cdot R}{P + R}, \tag{12}$$

where the notion of *precision* (P) is replaced with *accuracy* (i.e., the ratio between the number of documents correctly classified and the number of classified documents), and the notion of *recall* (R) is replaced with the *coverage* (i.e., the ration between the number of correctly classified documents and the number of documents that should be classified):

$$P = \frac{\#\text{ of correctly classified}}{\#\text{ of classified}} \quad R = \frac{\#\text{ of correctly classified}}{\#\text{ of documents in } \mathscr{D}}. \tag{13}$$

The evaluation of the quality of the datasets was performed with the introduction of a new measure referred to as *k-relative coverage*. In particular, *k-relative coverage* is a measure of the percentage of classes that were annotated with at least k examples. This measure provides an evaluation of the quality of the dataset because it highlights both the amount of information that the supervised learner could get from any class, and the a priori distribution of the classes. [9]

This second measure is useful to understand the quality of the classification hypothesis of the proposed bootstrapping model. Specifically, the higher this measure is for all k, the better the dataset generated by the *bootstrapping* models. An example of the *k-relative coverage* for all ks for a couple of taxonomies can be seen in Fig. 4.

The *bootstrapping* homogeneity was evaluated making a distinction between *macro* and *micro* measures. The former averages the measure over all nodes, while the latter computes the measure globally over all documents in the corpus. Specifically, the micro $F1$ measure combines total recall and total precision, providing the evidence of the global quality of the models. The macro $F1$ measure, on the contrary, combines the averages of local $F1$ for the classes. This allows for evaluating the degree of uniformity in the distribution of accuracy over all nodes—because the macro measures are independent from the prior probability of classes.

[9] In a uniform a priori distribution of classes with N documents and C classes, since there are $\frac{N}{C}$ documents per class, the *k-relative coverage* is equal to 100% for all $k \leqslant \frac{N}{C}$, and 0% elsewhere.

The evaluation of the proposed models, when considered part of the bootstrapping process, was performed using the standard information retrieval $F1$ measure as well. In particular, the models were evaluated assessing the quality of the dataset built by the models. This assessment is performed evaluating the accuracy of a very simple classifier (1-*NN*) on the different datasets built by the different models. Considering the motivation for these models, the higher the accuracy of the supervised model is, the better the dataset resulting from the bootstrapping phase, even if the unsupervised bootstrapping accuracy was low.

5. Evaluation of *TaxSOM* as stand-alone classification model

The *bootstrapping* task was conceived as a clustering problem, where the available knowledge can be used to constrain the training process in order to obtain a classifier. Clearly, the designed model cannot be compared with any supervised model, since there are no examples providing supervision. Moreover, it cannot be compared to any clustering algorithm that does not use the available knowledge. For this reason, two basic models were introduced, allowing for a fair evaluation of the results of *TaxSOM*: a *baseline* classifier and a *constrained K-means* algorithm.

5.1. The baseline approach

In order to get a fair evaluation of the proposed *TaxSOM* model, we designed a very simple algorithm to be used as touchstone. A straightforward classification technique to annotate a taxonomy with a set of unlabeled documents is to categorize documents according to the lexical information associated to the nodes in the taxonomy as done by Yang [31]. Specifically, a reference vector is built for each node, through the encoding of its labels. The documents are then associated to the node having the nearest reference vector (a standard prototype-based minimum error classifier). In the following, this simple class of keyword matching algorithms will be referred to as *baseline* categorization approach.

This classification method uses only lexical information, while topological information is neglected. The exploitation of only a part of the available knowledge implies poor responses. Moreover, any label can be used by various nodes in the taxonomy, and a document can contain many labels belonging to different nodes. This implies a high degree of ambiguity on the categorization process, and many documents need to be rejected.

This problem can be partially reduced using topological information as well. Specifically, topological knowledge can be exploited building codebooks through the encoding of all labels in the current node and in all ancestors, i.e., all nodes in the path from the root to the current node. In this way, each codebook encodes the local lexical information and part of the surrounding (contextual) lexical information. The idea is that the meaning of a node in a hierarchy of classes is a specialization of the meaning of its ancestors. Therefore, all keywords used to describe the meanings of ancestors can disambiguate the meaning of a node.

In this way, the rejection is strongly reduced but not completely eliminated. The amount of documents rejected by the *baseline* algorithm using the ancestor labels is usually halved with respect to the one that strictly uses the local information. Moreover, the algorithm gives better results with higher accuracy when contextual information is used. Hence, the *baseline* algorithm using

ancestor labels and rejection was used as a reference model to compare with *TaxSOM*. For a detailed discussion on the performance of all variations of the *baseline* approach refers to [2].

Notice that the *baseline* algorithm that only uses local information is the starting point for *TaxSOM* learning, i.e., the encoding of the codebooks of *TaxSOM* is equivalent to reference vectors of the *baseline* algorithm using local labels.

The comparison of this basic model versus *TaxSOM* allows the analysis of the influence on the model accuracy of the exploitation of the similarities between document content.

5.2. The clustering approach

Another way to approach the given task is with the clustering perspective. Differently from the *baseline* solution, clustering algorithms take advantage of document content. The main drawback, however, is the absence of a method to manage the prior knowledge.

One of the most known and used clustering algorithms is *K-means*. Here, we propose a variation of standard *K-means* that allows the use of both the document content and the knowledge embedded in the taxonomy. In particular, during training each codebook is computed using Eq. (6), as for standard *K-means* training. Then, the corresponding node labels [10] are encoded within the codebooks using Eq. (10). For this reason, the proposed model will be referred to as *constrained K-means*.

In the following, the results for *constrained K-means* using ancestors information are reported, because it proved to be more effective than the one using only local labels. For a detailed discussion on the performance of the two variations of *K-means* refers to [2].

Notice that the *constrained K-means* model that uses local information is a specific instance of *TaxSOM*, where the training procedure uses Gaussian neighborhood function with variance equal to zero. This assures that data is not propagated along the taxonomy. The comparison of *constrained K-means* versus *TaxSOM* allows the analysis of the influence on the classification accuracy of the contextual information gathered propagating the document content by means of Eq. (9).

5.3. Experimental results

Table 2 summarizes the experimental results of *baseline*, *constrained K-means*, and *TaxSOM* models, for all the taxonomies selected from the Google and LookSmart directories. For almost all measures and taxonomies, the *TaxSOM* model outperforms the *constrained K-means* algorithm, which, in turn, typically outperforms the *baseline* approach. This could be explained with the line of reasoning expressed in previous sections. Specifically, *constrained K-means* with contextual labeling starts its learning with the categorization result of *baseline*, and during training it improves the internal homogeneity of classes. In a similar way, *TaxSOM* starts the training algorithm using the results of *baseline* with local labels, but, instead of an explicit usage of a specific contextual information, it implicitly uses both labels and topology knowledge.

[10] Using either local labels or ancestor labels determines two variations of the *constrained K-means* model: one uses the local lexical information; the other one uses the contextual information.

Table 2
Stand-alone comparison of *TaxSOM* (TS) versus *baseline* (bas) and *K-means* (km)

	1-Relat. cover. (%)			Micro F1 (%)			Macro F1 (%)		
	bas	*km*	*TS*	*bas*	*km*	*TS*	*bas*	*km*	*TS*
Google									
Archaeology	83.61	84.43	90.16	27.60	26.00	31.56	58.16	55.98	60.75
Biology	41.41	42.91	80.82	21.75	21.73	37.48	34.01	34.72	71.52
Business	74.18	77.00	84.04	27.01	28.40	30.58	28.99	29.52	36.56
Cooking	68.99	71.66	84.42	19.01	21.90	38.15	28.79	32.59	48.98
Language	58.56	62.06	68.48	28.98	29.18	30.21	37.12	38.52	44.69
Neuro Disorders	88.10	90.00	83.33	47.33	45.34	46.69	54.75	53.17	57.30
News Media	93.10	96.55	100.00	34.29	33.70	39.34	33.97	34.60	39.24
Shopping Health	79.54	81.85	80.69	27.32	28.29	29.88	32.44	33.66	34.59
Technology	69.70	73.88	61.12	26.64	27.49	23.63	32.28	33.39	28.24
LookSmart									
Archaeology	66.67	66.67	75.64	26.63	26.65	28.97	31.93	31.85	36.89
Business Soft.	56.88	57.97	60.87	10.94	11.04	14.11	13.91	13.99	15.38
Common Lang.	70.71	70.71	74.29	18.25	18.91	20.11	22.79	22.53	24.56
Health Issues	39.20	40.91	43.56	13.80	14.05	14.81	15.25	15.65	16.00
Linguistics	57.68	57.68	61.13	23.27	23.08	22.30	29.45	29.13	30.65
Movies	85.29	88.24	88.24	26.70	25.81	35.19	34.70	34.29	38.77
Peripherals	61.62	62.63	66.67	19.67	19.90	32.16	20.29	20.64	25.09
Recipes	55.93	58.10	66.11	17.30	17.59	21.22	19.98	19.97	24.48
Videogames	66.67	67.15	70.74	35.40	35.20	36.88	38.11	37.54	40.85
Zoology	67.53	68.62	71.70	23.05	22.68	25.01	31.07	30.50	32.11

Results show that *TaxSOM* outperforms the other two approaches for almost all taxonomies.

Analysing the results in Table 2, it appears that the *1–relative coverage* of *TaxSOM* is almost always greater than *baseline* and *K-means*. This means that *TaxSOM* is able to correctly distribute the documents along (sometimes many) more nodes in the taxonomy than the other two classifiers. The result is that the probability of annotating all nodes with at least few good examples is increased. In particular, the *k-relative coverage* of the taxonomies annotated by *TaxSOM* is frequently higher than the results of the other two models for any k.

An example of such results is provided in Fig. 4, where two graphs show the *k-relative coverage* distributions for the annotations generated with the three models on two different taxonomies. The two graphs depict the *k-relative coverage* for the original annotations as well. This shows the upper bound for any learning algorithm for the given taxonomies. Remember that high *k-relative coverage* is good for the *bootstrapping* task, because it is the premise for an homogeneous assignment of the documents to the classes. Consequently, it is the premise for a highly accurate hierarchical supervised classifier.

The macro and micro $F1$ measures of Table 2 also provide the evidence that *TaxSOM* is usually better than *baseline* and *K-means*. It is worthwhile to notice that, for all models and for all taxonomies, the macro $F1$ measure is better that the micro $F1$ measure. This result shows that all approaches tend to uniformly distribute patterns over all concepts, increasing the correctness of the nodes with very few documents. This behavior is further emphasized by *TaxSOM*, which succeeds to increase the number of correctly classified documents for those nodes where *baseline* and

K-means fail. This is very good for the bootstrapping process, because it results in an increase of the average coverage and, although a lot of nodes have very few documents, the probability to annotate all nodes with at least some good examples is increased. This is very important for a subsequent supervised classification.

From the experiments we observed that the misclassified documents can be divided into two main classes: the class of documents whose terms match many category labels (both correct and wrong categories), and the class of documents whose terms do not match any of the labels in the correct category (but only the wrong ones). The explanation of the increased accuracy of *TaxSOM* can be explained by the fact that it is very effective in disambiguating the classification whenever terms occurring in documents match many category labels. On the contrary *baseline* and *K-means* have problems to find the correct labeling of such documents.

We observed that *TaxSOM*, as for *baseline* and *K-means*, is not effective in the classification of documents that do not include terms matching any of the labels of the correct category. This problem is related mainly to the dataset. Specifically, in the taxonomies, many nodes have very few documents and the documents usually have a small number of terms. Hence, the base of induction is very poor and the detection of category patterns can be very difficult.

6. Evaluation of *TaxSOM* as part of a complex process

In the previous section, the proposed models were evaluated as stand-alone classification algorithms. In addition, the models need to be evaluated as components of the *bootstrapping* process. To assess the quality of these models, considered a preliminary step of the process, the accuracy of a *k-Nearest Neighbor* classifier (*k-NN*) was evaluated on the annotated and cleaned taxonomies. This type of evaluation was proposed in [1] as well. In particular, because of the small number of documents per class, the *k-NN* classifier was applied with $k = 1$. To test the algorithm, a 10-fold cross validation method was used. Table 3 summarizes the classification performance of 1-*NN*, when the inductive base used is either the result of *baseline*, or the result of *TaxSOM*, or the original training set.

For each taxonomy, a 10-fold cross validation was performed. Specifically, 10 experiments were performed using 90% of documents to train the *bootstrapping* models. These documents will be referred to as the *bootstrapping set*. The wrongly classified documents were then removed, obtaining the *training set* for the supervised model. Finally, the remaining 10% of documents, referred to as *test set*, were classified with the 1-*NN* algorithm using the *training set* as inductive base. The results were then averaged over all 10 experiments. In addition, in order to find an upper bound to system performance, the 1-*NN* was also evaluated using the entire *bootstrapping set* as the inductive base.

The first important result, shown in Table 3, is that the classification task is rather difficult. In fact, the standard 1-*NN* classification using all possible data (*bootstrapping set*) does not achieve high accuracy. A clear result is that for both micro and for macro measures classification performance with the dataset produced by the *TaxSOM* annotation tends to outperform the classification performance obtained using the *baseline* labeling of documents.

Finally, another interesting result is that, different from the stand-alone evaluation of the models, the results of 1-*NN* with taxonomies of LookSmart are always better than the results with

Table 3
Comparison of the quality of training sets generated by *baseline* and *TaxSOM* compared versus the original *bootstrapping set*

	Micro F1			Macro F1		
	1-*NN* (*baseline*)	1-*NN* (*TaxSOM*)	1-*NN* (90%)	1-*NN* (*baseline*)	1-*NN* (*TaxSOM*)	1-*NN* (90%)
Google						
Archaeology	26.87	30.85	43.18	27.57	29.52	37.28
Biology	20.64	20.78	27.04	16.77	17.33	21.15
Business	35.26	35.37	37.89	27.90	28.77	29.42
Cooking	18.05	25.76	32.40	19.57	25.03	29.09
Language	24.93	16.93	29.66	16.98	15.97	21.75
Neuro Disorders	34.13	32.65	36.58	31.92	30.96	33.81
News Media	33.97	38.76	42.38	29.17	33.91	40.11
Shopping Health	28.76	31.91	33.86	27.09	29.50	30.03
Technology	24.64	22.46	28.78	22.48	20.57	25.19
LookSmart						
Archaeology	36.26	38.26	48.79	39.33	41.18	49.02
Business Soft.	51.07	52.95	54.55	54.52	54.99	54.53
Common Lang.	32.16	34.42	38.84	34.79	36.47	39.00
Health Issues	72.56	72.85	75.03	70.61	70.91	73.15
Linguistics	28.97	29.55	38.52	30.43	30.53	36.22
Movies	41.05	45.66	49.09	46.60	49.89	51.77
Peripherals	49.44	51.00	56.20	47.17	49.10	52.87
Recipes	56.11	57.76	59.50	55.77	56.39	56.69
Videogames	63.47	64.99	63.24	59.97	60.95	61.24
Zoology	31.79	32.61	36.02	33.75	34.35	36.08

Most of the time the dataset generated by *TaxSOM* is of higher quality than the one generated by *baseline*.

taxonomies of Google. This could be explained by the fact that the class descriptions (used in LookSmart, but not in Google) bias the feature selection process. This does not influence the training algorithm of the unsupervised models. On the contrary, the supervised classification gains quality. This because the feature selection is driven by the a priori knowledge inserted by human experts, which allows for a better description and discrimination of document content.

7. Related work

Supervised classifiers dealing with hierarchies of classes were recently proposed (see for example [7–9,12,13,18,20,26,27,29]). The common approach is to learn a supervised classifier for each node (i.e., category) of the taxonomy, and then to combine them according to different policies to obtain hierarchical classifiers. In any case, all different solutions share the same requirement of a minimum amount of labeled examples for each category. This requirement becomes critical when applied to complex domains where a huge number of labeled examples is required.

In literature, there are various works that aim at reducing the human effort during the *bootstrapping* process. Some works are based on the exploitation of both labeled and unlabeled examples (see for example [15,17,21,24]). Nevertheless, a first sample of classified documents is

always required. Moreover, these models do not take advantage of the relationships between the classes.

An alternative strategy to support the manual labeling of documents is active learning [10,25]. Active learning enables a document selection policy that reduces the end-user effort, focusing the labeling task on a restricted set of documents. The challenge in this case is to minimize the labeling effort without affecting the performance of the supervised classifier. However, as for the previous approaches, a minimum set of labeled examples is required.

The problem we are attacking can be conceived as a clustering task constrained by a given taxonomy. There are other initiatives dealing with the hierarchical unsupervised training, such as the idea promoted in [3], which, however, still needs a small set of labeled examples in advance. Moreover, the topology of the hierarchy is an output of the model.

The creation of a document classifier without any need for labeled example is a challenge also faced in [28]. The basic idea is to design classifiers that rely on groups of terms which are determined by extending the concept labels with related terms found in *WordNet* [14]. This approach, however, is not context sensitive in the sense that it does not exploit the relational information within the taxonomy.

An approach satisfying almost all the requirements of the task is proposed in [23]. In this work, a generative model based on naive Bayes is trained with EM to learn how to classify documents according to the node labels. Interestingly, the model uses a smoothing technique, referred to as *shrinkage*, that allows a partial propagation of information along the hierarchy as well. This propagation partially corresponds to the contextual processing of *TaxSOM*. The two models differ because the smoothing parameters in [23] are mainly driven by the content similarity, while smoothing in *TaxSOM* is mainly driven by the relationships between classes.

More specifically, both algorithms determine the class distributions according to the similarity of document content, and the starting point for the training algorithms is driven by the node labels. Moreover, the most probable keywords for any node are determined using both the knowledge of the documents currently classified in the corresponding class, and the information on the distribution of classes in the neighborhood. While *TaxSOM* exploits all the classes in the taxonomy, the model proposed in [23] only exploits ancestor classes, which are located on the path from the root node to current node. Since, however, the model proposed in [23] only classifies on the leaves, the distribution of keywords in interior classes is determined by a subsumption principle, [11] creating, in this way, "simulated" distributions for interior nodes.

The limitation of this approach is that documents are only classified into the leaves of the taxonomy. Besides this, the model uses only a part of the contextual information while classifying. Specifically, the shrinkage methodology allows the exploitation of "fake" distributions of all ancestors from the current node to the root.

The approach proposed in this paper pursues the same perspective of the work proposed in [23], but, instead of a probabilistic framework, the model takes advantage of some interesting features of SOMs [19]. SOMs have already been used to cluster documents into hierarchies of categories, and have been applied with success even to the web [7]. Nevertheless, with hierarchical SOMs the

[11] The distribution of a interior node is determined using all documents that are classified in any of the leaves of the sub-tree having the current node as root.

topology of the categories is a result of the learning process. In our task, on the contrary, the taxonomy is an input of the task and it influences the clustering algorithm during the learning.

8. Conclusions and future work

This paper proposes a process that helps a user create and manage a taxonomy of documents. Specifically, the system starts with an empty taxonomy and, using a set of unlabeled documents, generates a training dataset to be used for a further supervised classification task. The system can be described by three phases: a clustering phase that takes a taxonomy and a set of documents as input and returns a first hypothesis of annotation of the given taxonomy; a filtering step performed by a human expert that removes all wrongly classified documents from the annotated taxonomy, certifying the resulting dataset; and a final step where a supervised learning algorithm is trained with the remaining correct examples.

The main aim of this work was the exploration of the first phase of the process, which is neither a supervised nor an unsupervised task. When using supervised models, in fact, the target classes are known in advance, and labeled examples are provided for each target class. On the contrary, all unsupervised models are based on the assumption that nothing is known and only the similarity of patterns is used to learn a proper organization of classes. For this reason, the *bootstrapping* task could be referred to as a special kind of *supervised clustering* task, since the class descriptions and their organization are known, while examples are unlabeled.

Various methods are proposed to overcome the bootstrapping problem, such as the standard prototype-based classifier referred to as *baseline*, and the *constrained K-means* approach that, starting from a *baseline* result, performs a constrained clustering using document similarities. Above all, however, the *TaxSOM* model improves *baseline* and *K-means* by implicitly using the knowledge on the topology of the classes relationships.

TaxSOM showed better results than the other two methods, by analysing both its behavior as a stand-alone model, and the results of the whole process encapsulating the *bootstrapping* models. There are many reasons for such a result, and part of them were previously discussed. However, we observed some circumstances that can explain when *TaxSOM* behaves better than the other models. In particular, we observed that *TaxSOM* is significantly better than the other models when document descriptions are made of labels of wrong classes together with labels of the correct class, i.e., when keyword matching could be ambiguous.

8.1. Future works

Many documents do not contain the labels of the node to which they were originally assigned, and we have observed that *bootstrapping* algorithms tend to assign these documents to nodes having at least a label in the document. We are trying to tackle this problem by adjusting the lexical constraints that, in some cases, seem to be too strong.

An ongoing work is a further analysis of the *bootstrapping* process on web documents downloaded from Internet. In this task, many documents are retrieved by a simple query, and some documents cannot be related to the topics in the taxonomy. Such documents clearly should be

rejected, both during the preliminary classification and during the supervised classification. In this case, filtering according to the taxonomy domain needs to be performed.

Notice that in Eqs. (7) and (9) the propagation of information is homogeneous in all directions, i.e., the smoothing operator equally weights the information coming both from ancestors and from descendants. This because the *TaxSOM* topology is a graph with undirected edges. We experimented propagation schemes more respectful of the parent–child relationships, obtaining poor results, however. We plan to further explore different propagation schemes.

Appendix A. Google taxonomies

Archaeology	Science/Social Sciences/Archeology
Biology	Science/Biology
Business	Business/Business and Services
Cooking	Home/Cooking
Language	Science/Social Sciences/Language and Linguistics
Neuro disorders	Health/Conditions and Diseases/Neurological Disorders
News media	News/Media
Shopping health	Shopping/Health
Technology	Science/Technology.

Appendix B. LookSmart taxonomies

Archaeology	Science & Health/Social Science/Archaeology
Business Soft.	Computing/Sales/Software by Type/Business Software
Common Lang.	Computing/Computer Science/Programming/Ccommon Languages
Health issues	Science & Health/Health/Reference & News/Health Issues
Linguistics	Science & Health/Social Science/Linguistics
Movies	Entertainment/Movies/Reviews & News
Peripherals	Computing/Hardware/Peripherals
Recipes	Hobbies & Interests/Food Wine/Recipes
Videogames	Computing/Software/Software by Type/Computer & Video Games/Games/Games by Genre
Zoology	Science & Health/Biology/Zoology.

References

[1] G. Adami, P. Avesani, D. Sona, Bootstrapping for hierarchical document classification, in: Proc. of CIKM-03, 12th ACM Int. Conf. on Information and Knowledge Management, ACM Press, New York, 2003, pp. 295–302.
[2] G. Adami, P. Avesani, D. Sona, Clustering documents in a web directory, in: Proc. of WIDM-03, Fifth ACM Int. Workshop on Web Information and Data Management, ACM Press, New York, 2003, pp. 66–73.

[3] C. Aggarwal, S. Gates, P. Yu, On the merits of building categorization systems by supervised clustering, in: Proc. of KDD-99, Fifth ACM Int. Conf. on Knowledge Discovery and Data Mining, 1999, pp. 352–356.
[4] P. Avesani, C. Girardi, N. Polettini, D. Sona, TaxE: a testbed for hierarchical document classifiers. Technical Report T04-04-02, ITC-IRST, 2004. Available from: <http://www.sra.itc.it>.
[5] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999.
[6] M. Bonifacio, P. Bouquet, P. Traverso, Enabling distributed knowledge management managerial and technological implications, Informatik/Informatique 3 (1) (2002).
[7] M. Ceci, D. Malerba, Hierarchical classification of html documents with webclassii, in: Proc. of the 25th European Conf. on Information Retrieval (ECIR'03), Lecture Notes in Computer Science, vol. 2633, 2003, pp. 57–72.
[8] S. Chakrabarti, B. Dom, R. Agrawal, P. Raghavan, Using taxonomy, discriminants, and signatures for navigating in text databases, in: M. Jarke, M. Carey, K. Dittrich, F. Lochovsky, P. Loucopoulos, M.A. Jeusfeld (Eds.), VLDB'97, Proc. of 23rd Int. Conf. on Very Large Data Bases, Morgan Kaufmann, 1997, pp. 446–455.
[9] C. Cheng, J. Tang, A. Fu, I. King, Hierarchical classification of documents with error control, in: PAKDD 2001—Proc. of 5th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 2035, 2001, pp. 433–443.
[10] D. Cohn, Z. Ghahramani, M. Jordan, Active learning with statistical models, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), Advances in Neural Information Processing Systems, vol. 7, The MIT Press, 1995, pp. 705–712.
[11] Open directory project.
[12] H. Doan, P. Domingos, A. Halevy, Learning to match the schemas of data sources: A multistrategy approach, Machine Learning 50 (2003) 279–301.
[13] S. Dumais, H. Chen, Hierarchical classification of web document, in: Proc. of the 23rd ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR'00), 2000.
[14] C. Fellbaum, WordNet: An Electronic Lexical Database, The MIT Press, 1998.
[15] S. Goldman, Y. Zhou, Enhancing supervised learning with unlabeled data, in: Proc. 17th Int. Conf. on Machine Learning, 2000, pp. 327–334.
[16] Google directory.
[17] B. Jeon, D. Landgrebe, Partially supervised classification using weighted unsupervised clustering, IEEE Transactions on Geoscience and Remote Sensing 37 (2) (1999) 1073–1079.
[18] M. Jordan, R. Jacobs, Hierarchical mixtures of experts and the em algorithm, Neural Computation 6 (1994) 181–214.
[19] T. Kohonen, Self-Organizing MapsSeries in Information Sciences, vol. 30, Springer, Berlin, 2001.
[20] D. Koller, M. Sahami, Hierarchically classifying documents using very few words, in: D. Fisher (Ed.), ICML 1997, Proc of the 14th Int. Conf. on Machine Learning, 1997, pp. 170–178.
[21] B. Liu, W. Lee, P. Yu, X. Li, Partially supervised classification of text documents, in: Proc. 19th Intl. Conf. on Machine Learning, 2002, pp. 387–394.
[22] Looksmart directory.
[23] A. McCallum, K. Nigam, Text classification by bootstrapping with keywords, in: ACL99—Workshop for Unsupervised Learning in Natural Language Processing, 1999.
[24] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, Learning to classify text from labeled and unlabeled documents, in: Proc. of AAAI-98, 15th Conf. of the American Association for Artificial Intelligence, Madison, US, 1998, pp. 792–799.
[25] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2001, pp. 441–448.
[26] M. Ruiz, P. Srinivasan, Hierarchical text categorization using neural networks, Information Retrieval 5 (1) (2002) 87–118.
[27] A. Sun, E. Lim, Hierarchical text classification and evaluation, in: N. Cercone, T. Lin, X. Wu (Eds.), ICDM 2001—Proc. of the 2001 IEEE Int. Conf. on Data Mining, IEEE Computer Society, 2001, pp. 521–528.
[28] L. Ureña-López, M. Buenaga, J. Gómez, Integrating linguistic resources in TC through WSD, Computers and the Humanities 35 (2) (2001) 215–230.
[29] K. Wang, S. Zhou, S. Liew, Building hierarchical classifiers using class proximity, in: Proc. of the 25th VLDB Conference, 1999.

[30] Yahoo directory.
[31] Y. Yang, Expert network: effective and efficient learning from human decisions in text categorization and retrieval, in: Proc. of the 17th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, 1994, pp. 13–22.

Giordano Adami is a research assistant in the Automated Reasoning System Division of the Institute for the Scientific and Technological Research (ITC-IRST). He received a master degree in Computer Engineering from the University of Bologna, Italy. He is currently working at the TaxSOM project as software developer of the unsupervised classifier of documents in taxonomies. His past activities at ITC-IRST included the collaboration to a technology transfer project related to the development of custom tools for automating the development of controllers of industrial conditioning plants, and to a feasibility study of a peer-to-peer approach to distribute knowledge in large environments.

Paolo Avesani received his doctoral degree in Information Science from the University of Milan in 1988. Since 1989 he has been a research scientist in the Automated Reasoning System division at IRST. His research interests include case-based reasoning, machine learning, information retrieval and decision making. He is part of the ECCBR and ICCBR program committees and of the RIIA editorial board. He is a member of AAAI and AIIA.

Diego Sona received the Laurea degree in computer science from the University of Pisa, Italy, in 1996, and the Ph.D. degree from the University of Pisa in 2002. He is currently a research scientist at the Automated Reasoning Systems division of the Institute for Scientific and Technological Research (ITC-IRST), Trento, Italy. His research interests include neural networks, pattern recognition, information retrieval, relational machine learning and web mining.