Clustering Documents in a Web Directory

Giordano Adami
ITC-irst
via Sommarive 18
38050 Povo, Italy
gioadami@itc.it

Paolo Avesani
ITC-irst
via Sommarive 18
38050 Povo, Italy
avesani@itc.it

Diego Sona
ITC-irst
via Sommarive 18
38050 Povo, Italy
sona@itc.it

ABSTRACT

Hierarchical categorization of documents is a task receiving growing interest due to the widespread proliferation of topic hierarchies for text documents. The worst problem of hierarchical supervised classifiers is their high demand in terms of labeled examples, whose amount is related to the number of topics in the taxonomy. Hence, *bootstrapping* a huge hierarchy with a proper set of labeled examples is a critical issue. In this paper, we propose some solutions for the *bootstrapping* problem, implicitly or explicitly using a taxonomy definition: a *baseline* approach where documents are classified according to class labels, and two clustering approaches, where training is constrained by the *a-priori* knowledge of the taxonomy structure, both at terminological and topological level. 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 Web directory.

Categories and Subject Descriptors

F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*Self-modifying machines*; H.3 [**Information Storage and Retrieval**]: Miscellaneous; I.2.6 [**Artificial Intelligence**]: Learning; I.5.3 [**Pattern Recognition**]: Clustering; I.5.4 [**Pattern Recognition**]: Applications—*Text Processing*

General Terms

Algorithms, Experimentation.

Keywords

Web directories, TaxSOM, constrained clustering, k-means, taxonomy bootstrapping process, text categorization, knowledge management, digital libraries.

WIDM'03, November 7–8, 2003, New Orleans, Louisiana, USA.

1. INTRODUCTION

Hierarchical categorization of documents is a task receiving growing interest in information retrieval and machine learning communities, due to the widespread proliferation of topic hierarchies for text documents. A typical scenario is what we refer to as Web directories task, where the most relevant web pages are classified with respect to a predefined set of categories organized into hierarchies. Google, Yahoo and LookSmart are well known examples of such hierarchical organization. This categorization approach is strategic within company Intranets too, because knowledge management platforms very often support the hierarchical organization of information [4].

Ideally, the exploitation of this hierarchical organization provides the opportunity for the development of accurate classifiers, able to take advantage of the relationships between categories. Actually, some hierarchical classifiers dealing with this new challenge were recently proposed (see for example [2, 5, 6, 7, 8, 9, 10, 13, 19, 20]). All the proposed models rely on a supervised learning strategy, that allows to build a classifier starting from a set of labeled examples.

For any of the proposed supervised models, gathering a proper set of labeled examples[1], a process usually known in literature as *bootstrapping*[2], is a critical issue whenever applied to complex domains with many classes. This because the number of labeled examples needed for training is related to the number of categories. It is a matter of fact that the labels are usually provided by a person, hence, the creation of these sets of labeled data could be extremely tedious and expensive. Actually, for any document labeling, the domain expert needs to understand the semantics of the document content, and she also has to take a decision on the most appropriate labeling for the document, decision that is maybe taken while "navigating" the taxonomy of classes.

This paper addresses the *bootstrapping* problem for hierarchies of classes, aiming at the development of a supporting tool that allows to reduce the human effort required while annotating a taxonomy with examples. Specifically, we conceive *bootstrapping* within the Web directories domain as a three steps process: in the first step a preliminary filtering process produces a set of candidate documents from the

[1]A data set could be considered "proper" when the supervised learning algorithm can use as many labeled examples as needed in order to obtain a good generalization.
[2]The term *bootstrapping* is not related to the sampling theory but to a more general concept, i.e. *"to promote or develop by initiative and effort with little or no assistance"* (Merriam-Webster dictionary).

Web; in the second step a preliminary hypothesis on the classification of the candidate documents with respect to a given taxonomy is produced; and in the third step the misclassified documents are manually discarded by the domain expert. The assumption is that, having a good classification hypothesis for the data-set, it should be easier for a domain expert to detect wrong documents rather than, in a "constructive" way, to decide when and how to classify positive examples.

In this paper we focus our attention on the second step of the *bootstrapping* process. Specifically, we propose various solutions that provide a preliminary categorization hypothesis for a set of candidate documents, given a taxonomy definition, i.e. the class relations and their syntactic description (labels). It is worthwhile to remark that, we are clearly interested in a preliminary categorization with high accuracy but, since we are searching for a "proper" annotation of a given taxonomy, we also search for a tool obtaining a good coverage of the taxonomy concepts, i.e. all nodes of the taxonomy should be annotated by a "proper" number of labeled examples. A good coverage allows to have at least a small set of good examples for each node, increasing the probability for a good generalization of a supervised learning algorithm.

In literature there are various works aiming at the reduction of the human effort during the *bootstrapping* process. Some works are based on the exploitation of both labeled and unlabeled examples (see for example [11, 18], or a slight variation referred to as partially supervised classification [12, 16]). Nevertheless, a first sample of classified documents is always required. Moreover, these models are devised for non-hierarchical sets of classes.

The problem we are attacking is a clustering task constrained by a given taxonomy. There are other initiatives dealing with the hierarchical unsupervised training, such as the idea of hierarchical supervised clustering promoted in [1], which however still needs a small set of labeled examples in advance, and the model autonomously determines the taxonomy. An approach satisfying almost all task requirements is proposed in [17]. In this work, a taxonomy is bootstrapped using the set of class labels and their topological relations encoded using the concept of shrinkage. This approach, however, limits the documents classification only to the leaves of the taxonomy, while the Web directories task needs the placement of documents into interior nodes. Besides this, the model only uses a part of the contextual information while classifying. Specifically, the shrinkage methodology, used to encode labels, only uses information associated to nodes along the path from the leaves to the root.

The solutions proposed in the following can be divided into two approaches. In the first approach, to which we refer to as *baseline*, documents are classified on the basis of the node labels. The second approach on the contrary, is based on clustering models, where learning is constrained by the knowledge associated to the taxonomy definition. In particular, the two proposed clustering models, are based on the notion of *K-means* and Self-Organizing Maps (*SOM*s) [14]. The second, in particular, is an unsupervised learning technique used to cluster unlabeled data in a topologically related map of concepts. Such a technique was also applied to the Web [15] to obtain a hierarchical categorization of the Internet contents. In this case, however, the hierarchy of concepts is the result of a learning process instead of being the input to the process as required by the task.

directory: cooking/soups and stews/fish and seafood
labels: fish, seafood

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

uri: 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: snack, collection, recipe, healthy, savoury, vegetable, sweet

Figure 1: Documents contained in a node of directory *cooking*. All lemmata are used to create the vocabulary and to encode documents. Notice that document encodings are of very few keywords.

In Section 2 we introduce a real-world setting for the *bootstrapping* problem, providing details on the data and the measures used for the empirical evaluation of the model. Section 3 describes a *baseline* solution, which only exploits the lexical information of the taxonomy. Section 4 introduces a modified *K-means* algorithm constrained by the concept labels. Finally, Section 5 describes the *TaxSOM* model, which directly takes advantage both from the relationships of the category, the labels, and the set of candidate documents. Some experimental results are also reported.

2. TASK DEFINITION

The core of the *bootstrapping* process can be formalized as a function taking a taxonomy and a set of documents as input, and returning an association between documents and nodes of the taxonomy.

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

Nodes are the categories while edges describe the relationships between categories. A further assumption is that labels (L_N) are taken from the lexicon of natural language.

2.1 The Datasets

Web directories are meaningful example of taxonomy, hence, for our experiments we used a set of taxonomies extracted from the Google Web directory. By default, the unlabeled edges between classes in the hierarchy are "*is-a*" relationships. Without loosing generality, a further type of relationship explicitly labeled as "*related-categories*" was not used.

The taxonomies selected from Google directory were chosen according to common characteristics (see Table 1): hundreds of nodes, thousands of documents, no empty nodes (i.e. no nodes without documents), and almost all nodes marked with labels having a lexical meaning (e.g. without single character labels). The documents are URI descriptions, built using the web site title and the short summary given by the directory maintainers (usually a text of few tens up to hundred words), see the example in Figure 1.

Documents and node labels were cleaned by removing all stop-words (articles, conjunctions, and prepositions) and by transforming all remaining words into lemmata (stemming). Then, due to the usually small number of words repetition in a document description (see the sets of lemmata in the examples of Fig. 1), documents were encoded as binary vectors, indicating the presence or absence of the corresponding keywords in the document. The space of the encoding vector, i.e. the vocabulary, was separately determined for each taxonomy by a process of feature selection based on the notion of Shannon Entropy[3].

In particular, let L_N and L_D be the set of keywords appearing as labels of nodes in a given taxonomy and as words of documents in a given dataset respectively. The process of feature selection reduces the set L_D to a subset L'_D of about 500 keywords keeping the words with the highest entropy. Then a reference vocabulary $\mathcal{V}$ is built making the union of the two sets L'_D and L_N. In this way, the persistence of labels within the reference vocabulary is guaranteed.

In Table 1 are illustrated some statistics of the selected taxonomies after the preprocessing phase. The first section (Taxonomy) shows the features characterizing the subtaxonomies used for testing models. The second section (Vocabulary) illustrates the vocabulary dimensions, which are made-up of both the labels of nodes and the selected keywords of documents. Finally, the third section (Documents) gives a flavor of the problem complexity. In particular, the row "Labels in Docs" describes the occurrence of labels denoting a category in the documents corpora. The row "Contextual Labels in Docs" describes the percentage of documents that contain at least one label in the path from the root to the node where the document appear. Finally, the row "Local Labels in Docs" is the percentage of documents that contain at least one of the labels denoting the category in which they occur.

Notice that the percentages of labels in documents are rather high (mostly near 100%), while the "contextual" and "local" labels occurrences are significantly lower. This causes high ambiguity when trying to classify documents just looking at the labels occurrences, which in turn produces high rejection rates (see the *baseline* model below).

2.2 Evaluation of the Models

The effectiveness of the proposed bootstrapping solutions was assessed using two standard *Information Retrieval* measures: *precision* and *recall*. There is a trade-off between recall and precision due to *rejection*, thus it could be very difficult to assess the models. Therefore, for the sake of analysis and exposition we adopted the $F1$ measure [3], which combines precision and recall of a model on a given dataset. Moreover, since we are interested in the evaluation of the homogeneity of the proposed solution, we make the distinction between *macro* and *micro* accuracy. The former averages the measure over the nodes, while the latter computes the measure globally over all documents in the corpora.

Specifically, the micro $F1$ measure, that combines total recall and total accuracy, gives an idea of the global quality of the models. The macro $F1$ measure, instead, combines the average of the class recall and the average of the class accuracy. In this way, it is possible to evaluate the degree of uniformity in the distribution over all nodes of the documents that were correctly classified.

[3]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.

3. THE BASELINE APPROACH

A straightforward classification technique to annotate a taxonomy with a set of unlabeled documents is to simply categorize documents according to the lexical information (labels) associated to the nodes in the taxonomy as done by Yang [21]. Specifically, for each node a codebook (a reference vector also referred to as seed) is built through the encoding of its labels, and the documents are then associated to the node having the nearest codebook (a classic prototype-based minimum error classifier). In the following, we refer to this simple class of categorization algorithms as *baseline* approach.

This classification method only uses the lexical information, while the 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 an high degree of ambiguity on the categorization process and many documents need to be rejected, unless adopting a probabilistic classification scheme among ambiguous classes. The choice among rejection and pseudo-random assignment is driven by the trade-off between a strong reduction of the number of classified documents (high rejection) and an increase of the nodes coverage (i.e. recall), clearly accompanied by a decrease of correctness.

These problems are partially reduced by using both lexical and topological information. Specifically, the topological knowledge could be exploited building codebooks through the encoding of all node labels on 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. With this approach, the rejection problem is reduced but not completely eliminated.

As depicted in Table 2, the amount of documents rejected by the *baseline* algorithm using the context labels is usually halved with respect to the one that strictly uses the local information. Moreover, micro and macro $F1$ measures show that, for most of the tested taxonomies, the algorithm using the contextual information carry out better results than the one only using local information, both with and without rejection. The adoption of the rejection criterion does not notably change the model performance, since the model accuracy is increased at the cost of a reduction of classified documents.

In the following, we prefer to adopt the algorithm with best $F1$ measures as a reference model, to be used for comparisons with other algorithms. In particular, we observed that for most taxonomies the best *baseline* algorithm is the one that uses the contextual information and that rejects the ambiguous documents.

4. THE CLUSTERING APPROACH

The proposed *baseline* approach has important limitations. First of all, it only considers the taxonomy labels, without exploiting any other type of information, such as for example the relationships among document contents. Moreover, it does not distribute the documents homogeneously in all classes. Therefore, the resulting taxonomy annota-

	Directories							
	Arch	Biol	Buss	Cook	Lang	Neur	News	Heal
Taxonomy								
Max tree depth	5	11	5	7	8	5	4	6
Total nodes	125	1610	213	674	515	210	29	259
Total Documents	1163	9202	7180	16124	4125	2446	517	8683
Vocabulary								
Label words	210	1671	238	595	501	230	36	312
Vocabulary size	729	2140	682	869	883	644	561	702
Average words/docs	10.80	8.61	8.88	7.86	8.91	9.63	10.34	8.23
Documents								
Labels in Docs (%)	94	100	99	100	100	88	98	99
Contextual Labels in Docs (%)	72	73	85	89	90	70	89	77
Local Labels in Docs (%)	45	60	62	75	64	51	70	58

Arch = science>social sciences>archeology
Biol = science>biology
Buss = business>business and services
Cook = home>cooking
Lang = science>social sciences>language and linguistics
Neur = health>conditions and diseases>neurological disorders
News = news>media
Heal = shopping>health

Table 1: Some taxonomies statistics. First two sections are self-explanatory. Section "Documents" provides the occurrence of labels in the documents: "Label in Docs" is the percentage of documents that include at least one label in L_N, "Contextual Labels in Docs" is the percentage of documents that include at least one label of the node in the path from the root to the current node, "Local Labels in Docs" is the percentage of documents that include at least one label of the current node.

tion could have a low macro recall (i.e., many nodes can be empty), hence, a low coverage.

An alternative way to look at the problem is the clustering perspective. Clustering algorithms allow to exploit the information coming from unlabeled examples. Through an unsupervised approach, the documents could be organized according to their contents. The main drawback, however, is the absence of a method to control the exploitation of the *a priori* knowledge, unless disposing of a small set of labeled examples. Many clustering algorithms were devised in the past. One of the most known is the *K-means* algorithm. In this work we propose a small variant of standard *K-means*, which allows to use both the document contents and the knowledge brought by the taxonomy specification.

A simple stratagem used to accelerate the clustering processes is to start the algorithms with codebooks properly initialized [14]. In particular, a good initialization of codebooks for the current task could be one of those proposed for the *baseline* algorithms, i.e. encoding locals or contextual labels into the class centroids. We observed that, with this approach, the model quickly converges to a solution, which however is of poor quality, in the sense that the nodes coverage and the classification accuracy are very low. Actually, the algorithm starts the clustering process with a "nearly-good" documents classification (the same as for the *baseline* algorithms), but, during "training", the organization of documents changes according to criteria different from the "classification" purposes. The main problem is the lack of constraints in the assignment of documents to codebooks.

To minimize the inter-category variances while trying to preserve a good classification accuracy, we modified the *K-means* algorithm to be constantly constrained by the labels during all training steps. Specifically, for any iteration of the learning models each codebook is computed as in standard *K-means* training algorithm and then the corresponding local or contextual labels are encoded within the codebook. Hence, we obtain two variants of the *K-means* algorithm, one that uses the local lexical information, and the other that uses the contextual information.

In the current implementation we adopted the standard *K-means* training algorithm, computing the class centroids c_i as the average of documents classified in the class:

$$\boldsymbol{c}_i = \frac{\sum_{\boldsymbol{x} \in D_i} \boldsymbol{x}}{|D_i|} \tag{1}$$

where D_i is the *Voronoi set* of class i (i.e., the set of documents classified in class i). Then the centroids are updated also encoding the local or the contextual labels as follows:

$$\boldsymbol{c}'_i = f(\boldsymbol{c}_i, L_N(i)) \tag{2}$$

where $L_N(i)$ is the set of labels of node i, $\boldsymbol{c}_i$ is the starting centroid, $\boldsymbol{c}'_i$ is the resulting centroid also encoding the labels, and $f()$ is a function describing how labels are encoded within the codebooks. In the current implementation the function $f()$ simply forces to be 1 all those elements of the centroid corresponding to the labels:

$$\boldsymbol{c}' = f(\boldsymbol{c}, L) \quad \text{where} \quad c'_j = \begin{cases} 1 & \text{if } \mathcal{V}(j) \in L \\ c_j & \text{otherwise} \end{cases} \tag{3}$$

where $\mathcal{V}(i)$ is a function returning the i-th keyword in the vocabulary[4], and c_j (respectively c'_j) is the j-th element of vector $\boldsymbol{c}$ (respectively $\boldsymbol{c}'$).

In Table 3 are reported the results for the two versions of the "constrained" *K-means* algorithms. We omitted the

[4]Since in the current task the documents are binary vectors, all elements of the codebook vectors are real number in the interval $[0, 1]$.

	Directories							
	Arch	Biol	Buss	Cook	Lang	Neur	News	Heal
Rejected Docs (%)								
Contextual with Rejection	26.83	22.66	21.53	34.66	20.34	21.38	29.01	23.46
Local with R.	43.94	55.31	43.52	81.41	57.92	39.04	45.07	50.16
Micro $F1$ (%)								
Contextual with Rejection	28.30	21.72	26.36	19.02	28.50	47.29	33.71	27.33
Contextual	26.05	21.16	26.39	20.96	29.21	44.77	32.30	27.27
Local with Rejection	27.11	26.45	18.53	10.19	20.64	39.01	29.96	21.58
Local	26.31	28.11	20.38	16.73	21.67	36.35	30.17	22.45
Macro F1 (%)								
Contextual with Rejection	58.80	33.97	28.88	28.83	36.34	54.72	34.30	32.48
Contextual	49.03	33.61	28.84	31.90	38.72	53.87	31.43	32.02
Local with Rejection	56.48	39.25	16.92	17.18	28.86	49.12	30.38	24.23
Local	48.71	55.33	19.13	28.87	33.50	45.39	29.98	27.57

Table 2: *Baseline* results. For most taxonomies the best *baseline* algorithm is the one using contextual information and rejecting ambiguous documents.

results of "unconstrained" *K-means* algorithms due to their not meaningful (extremely poor) results. The algorithms were tested rejecting documents whenever there was an ambiguous categorization, but, after some training training, no more rejection was observed. Considering the micro $F1$ measure, *K-means* with contextual labeling is usually more effective than the one only using local labels. The macro $F1$ measures, on the contrary, does not provide a clear evidence of the model that get better effectiveness. These empirical results prove that the *K-means* model using contextual labeling is not effective in exploiting the topology information.

5. EXPLICITLY USING THE TOPOLOGY

From the previous two sections, it emerges that both *baseline* and *K-means* approaches seem to be able to partially classify the documents, using both the lexical information associated to the nodes and the topological information described by the taxonomy structure. Nevertheless, the performance improvement due to the topology knowledge is less evident than what can be expected. This because, the class relationships are not explicitly and entirely exploited. Actually, their implicit use depends on the fact that only a part of the contextual lexical information is encoded into the models.

The problem can be overcame by using a learning model able to explicitly exploit the class relationships. For this reason we propose a new model, which is derived from the Self-Organizing Map model (*SOM*) [14]. This new model, referred to as *Taxonomic Self-Organizing Map* (*TaxSOM*), is able to cluster data according to a given taxonomy, explicitly using information coming both from the topology and from the class labels. The idea behind the model is very simple. A *SOM* can be seen as a collection of classes related each other according to a fixed topology. These topological relationships have a strong impact on *SOM*s training and behavior, indicating that this property can be very useful for the proposed task.

A *SOM* consists of k computational units located on a regular low-dimensional grid $\mathcal{A}$, usually planar with rectangular or hexagonal connection schemes. Each unit is described by both a position index in the lattice, and a codebook vector $\boldsymbol{w}_i = [w_1, ..., w_m]$, which is a cluster centroid in the input space. *SOM*s are trained by alternating between a *competitive* and a *cooperative* phase for each presented input pattern. During *competitive* stage, the codebook most similar to the input vector $\boldsymbol{x}$ is chosen as the *winner* unit. In the *cooperative* stage, all codebooks are moved closer to the input vector, with a learning rate proportional to the inverse of their topological distance from the winner unit:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \eta(t) h_{i,i^*}(t)[\boldsymbol{x} - \boldsymbol{w}_i(t)] \qquad (4)$$

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:

$$h_{i,i^*}(t) = exp(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_{i^*}\|^2}{2\sigma^2(t)}) \qquad (5)$$

where $\sigma(t)$ is the function range (width of the neighborhood) decreasing in time, while $\boldsymbol{r}_i$ and $\boldsymbol{r}_{i^*}$ are the coordinates of respectively i and i^* on the discrete lattice $\mathcal{A}$.

5.1 The *TaxSOM Model*

The main property of *SOM*s is that similarity relations of patterns in the input space are mapped into similarity relations between codebooks, that is, documents belonging to similar concepts are mapped to the same unit (codebook) or to near units in the lattice.

Starting from this property, the idea is to model the topology of a *SOM* according to the taxonomy we are trying to annotate. Specifically, given a taxonomy $\mathcal{T}$, a *TaxSOM* is a self organizing map with the network topology equal to the taxonomy topology. Specifically, for each node in $\mathcal{T}$ there is a computational unit in *TaxSOM*, and for any directed edge connecting two nodes in $\mathcal{T}$ there is an undirected edge connecting the corresponding units in *TaxSOM* (see the example in Figure 2).

The conjecture is that, once a *TaxSOM* has been trained, the final codebooks configuration describes a clustered organization of documents, according to the desired topology relations. However, there is still an unresolved problem. The model, as it is, only exploits the topological organization of classes.

A simple way to also exploit the labeling knowledge is inherited from the previous modification of the *K-means* al-

	Directories							
	Arch	Biol	Buss	Cook	Lang	Neur	News	Shop
Micro $F1$ **(%)**								
Contextual	26.88	21.45	28.14	22.44	28.58	45.63	32.69	28.24
Local	27.02	29.31	23.45	21.34	25.75	41.99	32.69	26.19
Macro $F1$ **(%)**								
Contextual	55.78	34.72	29.82	32.74	37.64	53.37	33.77	34.04
Local	57.28	47.44	20.72	37.62	37.27	52.11	35.83	31.31

Table 3: $F1$ results for *K-means* algorithm using either local or contextual labels.

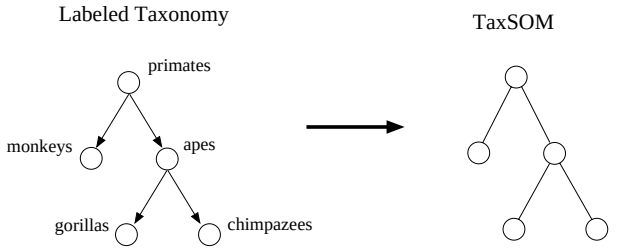

Figure 2: *TaxSOM* is a self-organizing map where units are connected according to the topology of taxonomy over which data must be clustered.

gorithm. Specifically, for any iteration of the model, each codebook is computed as in the standard *SOM* training algorithm, then, the corresponding local or contextual labels are encoded within the codebook. In this way, *TaxSOM* training is biased both by the knowledge of the concepts relations (taxonomy structure), and by the knowledge of the concepts descriptions (labels).

In the current implementation of *TaxSOM*, we adopted a batch training algorithm [14], in order to both reduce the computational cost of the simulations while addressing large data sets, and to obtain a smoothed convergence to a solution with low variance. With this approach, the algorithm requires a small number of iterations, since the *competitive* phase is performed on the whole data set before computing the *cooperative* step. In particular, the *cooperative* phase of the batch training algorithm is formulated using three steps. In the first step, for any node in the taxonomy, a centroid $\boldsymbol{x}_i(t)$ is computed as the average of the documents classified in the node. In the second step a smoothing procedure is carried out on the centroids obtaining a "temporary codebook":

$$\boldsymbol{w}_i(t+1) = \frac{\sum_j n_j(t) h_{i,j}(t) \boldsymbol{x}_j(t)}{\sum_j n_j(t) h_{i,j}(t)} \qquad (6)$$

where $n_j(t)$ is the number of data samples falling into the *Voronoi set* of unit j, and $h_{i,j}(t)$ is a neighborhood function similar to the one of Equation 5. The distance measure used by the neighborhood function $h_{i,j}$ in the *TaxSOM* model is computed as the shortest path between the two nodes i and j in the taxonomy. Finally, in the third step, the codebooks are computed encoding the labels into the "temporary codebooks". This operation is carried out by the function $f()$ described in Equations 2 and 3.

Notice that, if the models starts the training procedure using a variance of the Gaussian neighborhood function equal to zero, then there is not data propagation along the network. In this case, the *TaxSOM* model is equivalent to the above variation of the *K-means* model.

The current implementation of *TaxSOM* uses a variation of the batch-SOM training algorithm [14], which is "constrained" by the local labeling. This because the local labeling proved to be more effective than the global labeling. Our intuition is that the use of contextual labeling, added to the intrinsic ability of *TaxSOM* to use topological information, creates too strong constraints.

5.2 Experimental Results

Table 4 summarizes the experimental results for *baseline*, *K-means*, and *TaxSOM* models, on all the taxonomies selected from the Google directory.

For almost all measures and taxonomies the *TaxSOM* model overcome the "constrained" *K-means* algorithm, which in turn overcome the *baseline* approach. This could be explained by the line of reasoning expressed in previous sections. Specifically, "constrained" *K-means* with contextual labeling begins its learning algorithm with the categorization obtained by the *baseline* approach, and iterating it improves the internal homogeneity of classes. In the first iteration, *TaxSOM* starts in a way similar to *K-means* and *baseline*, but, instead of implicitly using a representation of part of the topology, it explicitly uses both labels and topology knowledge.

The explanation of the better behavior of *TaxSOM* is related to the error components. From the experiments, we observed that the misclassified documents can be divided into two main classes: the class of documents whose terms overlap many category labels (both the right category and wrong categories), and the class of documents whose terms don't overlap the right category label (but only the wrong ones). *TaxSOM* is very effective in resolving the ambiguity related to the occurrence of terms that overlap many categories. On the contrary *baseline* and *K-means* can not resolve the right assignment of the document to the proper node. In Biology, Cooking, and Languages taxonomies, the ambiguity due to the overlaps with many category labels is higher than in the remaining taxonomies, hence, *TaxSOM* gives better results.

TaxSOM is not effective when the documents do not include terms overlapping the label of the correct category. This problem is related mainly to the dataset: in the Google taxonomies many nodes have very few documents and the documents usually have a small number of terms. Moreover, a small percentage of documents contain the labels of the correct class. Hence, the base of induction for the experimented datasets is very poor and the detection of category patterns can be really difficult.

	Directories							
	Arch	Biol	Buss	Cook	Lang	Neur	News	Shop
Micro $F1$ (%)								
baseline	28.30	21.72	26.36	19.02	28.50	47.29	33.71	27.33
K-means	26.88	21.45	28.14	22.44	28.58	45.63	32.69	28.24
TaxSOM	31.56	37.71	30.22	32.21	30.21	47.05	39.34	29.85
Macro $F1$ (%)								
baseline	58.80	33.97	28.88	28.83	36.34	54.72	34.30	32.48
K-means	55.78	34.72	29.82	32.74	37.64	53.37	33.77	34.04
TaxSOM	61.21	71.82	35.98	45.16	44.84	56.26	40.04	34.57
Average Coverage								
baseline	2.36	1.10	7.93	3.76	2.05	4.92	5.14	8.09
K-means	2.56	1.22	9.46	5.34	2.29	5.31	5.83	9.47
TaxSOM	3.11	2.16	10.73	7.79	2.42	5.47	7.44	9.97

Table 4: *TaxSOM* with "node" labeling and rejection is compared with the *baseline* and *K-means* approaches. The assessment is performed using micro $F1$, macro $F1$, and the average coverage.

It is worthwhile to notice that, for all models and for all taxonomies, the micro F1 measure always overcome the macro F1 measure. This result proves 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 the *TaxSOM* model, which succeeds to increase the number of correct classified documents for those nodes where *baseline* and *K-means* fail. This is very good for the bootstrapping task because the average coverage is increased (see third part of Table 4) and, although there are a lot of nodes with very few documents, the probability to annotate all nodes by at least some good examples is increased. This is very important for a subsequent supervised classification, since it is the premise to obtain an homogeneous assignment of the documents to the nodes and, consequently, to obtain an highly accurate hierarchical supervised classifier.

6. CONCLUSION

In this paper we proposed a classification task which is neither supervised nor unsupervised. 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 similarity of patterns is used to learn a proper organization of classes. For this reason we could refer to the bootstrapping task as a special kind of "supervised clustering" task, since the target classes description and their organization are known, while examples are unlabeled.

We proposed various methods to overcome with the bootstrapping problem, such as the standard prototype-based classifier referred to as *baseline* approach, and the "constrained" *K-means* approach that, starting from a *baseline* result, performs a constrained data clustering. Above all, however, we proposed the *TaxSOM* model, which improves the *baseline* and *K-means* performance by explicitly including the taxonomy knowledge into the model.

The *TaxSOM* model showed better results than the other methods. There are many reasons for such results, 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 documents descriptions are made of labels of wrong classes together with labels of the correct class.

Notice that in Equation 6 the propagation of information is homogeneous in all directions, i.e. the update of a given centroid equally depends on 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 however poor results. We plan to further explore different propagation schemes.

A further investigation is concerned with the capability of TaxSOM to filter the unrelated documents. In this paper we considered the scenario where the candidate set of documents is related to the taxonomy. We will also consider the case where the candidate set also includes documents not being in any of the taxonomy topics.

7. REFERENCES

[1] C. Aggarwal, S. Gates, and P. Yu. On the merits of building categorization systems by supervised clustering. In *Proc. of KDD-99, 5th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 352–356, 1999.

[2] E.-P. L. Aixin Sun. Hierarchical text classification and evaluation. In N. Cercone, T. Lin, and X. Wu, editors, *ICDM 2001 - Proc. of the 2001 IEEE Int. Conf. on Data Mining*, pages 521–528. IEEE Computer Society, 2001.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] M. Bonifacio, P. Bouquet, and P. Traverso. Enabling distributed knowledge management. managerial and technological implications. *Informatik/Informatique*, 3(1), 2002.

[5] M. Ceci and D. Malerba. Hierarchical classification of html documents with webclassii. In *Proc. of the 25th European Conf. on Information Retrieval (ECIR'03)*, volume 2633 of *Lecture Notes in Computer Science*, pages 57–72, 2003.

[6] S. Chakrabarti, B. Dom, R. Agrawal, and P. Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In M. Jarke, M. Carey, K. Dittrich, F. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, editors,

VLDB'97, Proc. of 23rd Int. Conf. on Very Large Data Bases, pages 446–455. Morgan Kaufmann, 1997.

[7] C.H.Cheng, J. Tang, A. Wai-chee, and I. King. Hierarchical classification of documents with error control. In *PAKDD 2001 - Proc. of 5th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, volume 2035 of *Lecture Notes in Computer Science*, pages 433–443, 2001.

[8] M. S. D. Koller. Hierarchically classifying documents using very few words. In D. Fisher, editor, *ICML 1997, Proc of the 14th Int. Conf. on Machine Learning*, pages 170–178, 1997.

[9] H. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: A multistrategy approach. *Machine Learning*, 50:279–301, 2003.

[10] S. Dumais and 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.

[11] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proc. 17th Int. Conf. on Machine Learning*, pages 327–334, 2000.

[12] B. Jeon and D. Landgrebe. Partially supervised classification using weighted unsupervised clustering. *IEEE Trans. on Geoscience and Remote Sensing*, 37(2):1073–1079, 1999.

[13] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[14] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Series in Information Sciences*. Springer, Berlin, third extended edition, 2001.

[15] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Trans. on Neural Networks*, 11(3):574585, 2000.

[16] B. Liu, W. Lee, P. Yu, and X. Li. Partially supervised classification of text documents. In *Proc. 19th Intl. Conf. on Machine Learning*, pages 387–394, 2002.

[17] A. McCallum and K. Nigam. Text classification by bootstrapping with keywords. In *ACL99 - Workshop for Unsupervised Learning in Natural Language Processing*, 1999.

[18] K. Nigam, A. McCallum, S. Thrun, and 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*, pages 792–799, Madison, US, 1998.

[19] M. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Information Retrieval*, 5(1):87–118, 2002.

[20] K. Wang, S. Zhou, and S. Liew. Building hierarchical classifiers using class proximity. In *Proc. of the 25th VLDB Conference*, 1999.

[21] 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*, pages 13–22, 1994.