# Hierarchical Transductive Classification from Textual Data with Relevant Example Selection

Michelangelo Ceci and Pasqua Fabiana Lanotte

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
ceci@di.uniba.it; fabiana.lan@hotmail.it

**Abstract.** In many textual repositories, documents are organized in a hierarchy of categories to support a thematic search by browsing topics of interests. In this paper we present a novel approach for automatic classification of documents into a hierarchy of categories that works in the transductive setting and exploits relevant example selection. While the transductive learning setting permits to classify repositories where only few examples are labelled by exploiting information potentially conveyed by unlabelled data, relevant example selection permits to tame the complexity of the task and increase the rate of learning by focusing only on informative examples. Results on real world datasets are reported.

## 1 Introduction

Transductive learning is an inference mechanism adopted from several classification algorithms capable of exploiting, as in *semi-supervised learning*, information potentially conveyed by unlabelled data to better estimate the data distribution when making predictions. However, transductive learning differs from *semi-supervised learning* since, instead of learning a function to be used to make predictions on any possible example, it is only possible to make predictions for the given set of unlabeled data. This means that transductive learning needs no general hypothesis and appears to be an easier problem than both semi-supervised learning and classical inductive learning.

Several transductive learning methods have been proposed in the literature for classification tasks. They exploit SVMs ( [4]), k-NN classifiers ([9]) and even general classifiers ([10]). However, a common problem in this learning setting comes from the high dimensionality of unlabeled data and labeled data that have to be simultaneously analyzed during learning. In order to face this problem, two orthogonal directions can be exploited: the first direction aims at simplifying the classification process by considering that categories can be organized hierarchically. The second direction aims at simplifying the process by considering only a subset of examples for learning (relevant examples selection).

Both directions can be profitably pursued in the context of document categorization [16] that we consider in this paper. In fact, Hierarchical text categorization, that is, the process of automatically assigning one or more predefined

categories to text documents where the pre-defined categories are organized in a tree-like structure, has received increasing attention [6, 15]. From an information retrieval viewpoint, this hierarchical arrangement is essential when the number of categories is high, since thematic search is made easier by browsing topics of interests (as in Yahoo, Google Directory, MeSH, etc.).

This hierarchical structure of categories may help to simplify the classification process: while in flat classification a given example is assigned to a category on the basis of the output of one or a set of classifiers, in hierarchical classification the assignment of a document to a category can be done on the basis of the output of multiple sets of classifiers, which are associated to different levels of the hierarchy and distribute example among categories in a top-down way. The advantage of the hierarchical classification is that the problem is partitioned into smaller subproblems, each of which can be effectively and efficiently managed[3].

As for relevant example selection, in [1], the authors observed that there are at least three reasons for selecting examples to be used during the learning process: *i)* the learning process is computationally intensive, *ii)* the cost of manual labelling is high, *iii)* it is necessary to increase the rate of learning by focusing only on informative examples. In this context, all these motivations make relevant example selection particularly suited. Surprisingly, there is not much research on relevant example selection for text classification. The issue is mostly addressed either with the traditional statistical approach of sampling [21] or by more elaborate, but sometimes heuristic, approaches [20].

In this paper, we investigate the use of transductive learning by exploiting both hierarchical classification and relevant example selection. At this aim, we exploit a modified version of an inductive hierarchical learning framework that permits to classify examples (documents) in internal and leaf nodes of a hierarchy of categories. The learner is asked to take into account only a subset of the original documents. This way it is possible to speed up learning times without loosing in accuracy. Transductive learning exploits the Spectral Graph Transducer (SGT) [9], in the context of the hierarchical classification framework WebClass [3] originally designed for inductive classification.

## 2 Preliminaries

The problem we intend to solve can be formalized as follows:

Let $D$ be a set of documents and $\Psi : D \rightarrow Y$ be an unknown target function, whose range is a finite set $Y = \{C_1, C_2, \ldots, C_L\}$ where $\{C_1, C_2, \ldots, C_L\}$ are categories organized according to a tree-like structure such that $\forall i = 2, \ldots, L \quad \exists | \; j = 1, \ldots, L$, $i \neq j$ such that $C_i$ is a subcategory of $C_j$ ($C_1$ is the root category). Then, the transductive classification problem can be defined as follows:

*Given*: *i*) a training set $TS$ of pairs $(d_i, y_i)$ where $d_i$ represents a document and $y_i \in Y$ represents the class (label); *ii*) a working set $WS$ of unlabelled documents. *Find*: a prediction of the class value of each document in $WS$.

The learner receives full information (including labels) on the documents in $TS$ and partial information (without labels) on the documents in $WS$ and is required to predict the class values only of the examples in $WS$.

The hierarchical organization of categories adds additional sources of complexity to the transductive learning problem. First, documents can either be associated to the leaves of the hierarchy or to internal nodes. Second, the set of features selected to build a classifier can either be category specific or the same for all categories (corpus-based). Third, the classifier may or may not take into account the hierarchical relation between categories. Forth, a stopping criterion is required for hierarchical classification of new documents in non-leaf categories.

We face such complexity by resorting to solutions investigated in a previous work done on hierarchical classification in the classical inductive setting [3]. Those solutions have been implemented in the system WebClass. In WebClass, the search proceeds top-down from the root to the leaves according to a greedy strategy. When the document reaches an internal category $C$, it is represented on the basis of the feature set associated to $C$. The classifier of category $C$ returns a score for each direct subcategory. Score thresholds, which are automatically determined for all categories, are used to filter out the set of candidate subcategories. If the set is empty, then search is stopped, otherwise the subcategory corresponding to the highest score is selected and the (greedy) search recursively proceeds with that subcategory (if not leaf). The last crossed node in the hierarchy is returned as the candidate category for document classification (*single-category classification*). If the search stops at the root, then the document is considered *unclassified*. Each document is represented at decreasing levels of abstraction by considering features selected according to the $maxTF \times DF^2 \times ICF$ [3]. According to the definition of such measure, features tend to be more specific for lower level categories.

During learning, for each internal category $C$ of the hierarchy and for each document in $WS$ to be classified, the decision on which category $C'$ among the direct subcategories of $C$ is the most appropriate to receive the document has to be taken. In general, however, a document should not be necessarily passed down to a subcategory of $C$. This makes sense in the case that the document to be classified deals with a general rather than a specific topic, or in the case that the document to be classified belongs to a specific category that is not present in the hierarchy and it makes more sense to classify the document in the "general category" rather than in a wrong category. To support the classification of documents also in the internal categories of the hierarchy, it is necessary to compute the thresholds that represent the "minimal score" (returned by the classifier), such that a document can be considered to belong to a direct subcategory. More formally, let $\gamma_{C \to C'}(d)$ denote the score returned by the classifier associated to the internal category $C$ when the decision of classifying the document $d$ in the subcategory $C'$ is made. Thresholds are used to decide if a new testing document is characterized by a score that justifies the assignment of such a document to $C'$. Formally, a new document $d \in WS$ temporary assigned to a category $C$ will be passed down to a category $C'$ if $\gamma_{C \to C'}(d) > Th_C(C')$, where $Th_C(C')$ is

the score threshold. The algorithm for the automated determination of thresholds $Th_C(C')$ is based on a bottom-up strategy and minimizes a *tree distance* measure[3].

## 3 Relevant Example Selection

When working in the transductive setting, we do not distinguish between learning and classification steps. However, the hierarchical organization of categories requires a preliminary step during which thresholds are automatically identified. Later on, in a second stage, the transductive classification is performed. Indeed, the two phases are not completely independent each other since the algorithm for automatic threshold identification estimates thresholds on the basis of a simulation of the classification step on the training set. Relevant example selection is performed both in the automatic threshold determination and in the transductive classification task. In particular, while in automatic threshold determination relevant examples are determined from the training set $TS$ for each internal category $C$ of the hierarchy, in the classification case, both examples in $TS$ and examples in $WS$ are analyzed for each internal category $C$ of the hierarchy. Transductive classification of relevant examples in $WS$ is then extended to other examples in $WS$ by means of a K-NN label propagation.

For relevant example selection, we consider two different approaches: the first approach reduces the number of documents by exploiting clustering algorithms, while the second approach identifies and keeps only documents that are at the boundary of the class. Before describing how relevant documents are selected, we present details on their representation.

### 3.1 Document Representation

Document representation depends on a preprocessing step which aims at *i*) removing *stopwords*, such as articles, adverbs, prepositions and other frequent words; *ii*) determining equivalent stems (*stemming*) by means of Porter's algorithm for English texts [14]. After these preprocessing steps, documents are represented by means of a feature set which is determined on the basis of some statistics whose formalization is reported below. Let $C$ be an internal node in the hierarchy of categories, $C'$ a direct subcategory of $C$, $d$ a training document from $C'$, $w$ a token of a stemmed (non-stop)word in $d$, $TF_d(w)$ the *relative* frequency of $w$ in $d$, $Training(C) \subseteq TS$ the set of documents in $C$ and its subcategories, $TF_{C'}(w) = max_{d \in Training(C')} TF_d(w)$ the maximum value of $TF_d(w)$ on all training documents $d$ of category $C'$, $DF_{C'}(w) = \frac{|\{d \in Training(C') | \quad w \ occurs \ in \ d\}|}{|Training(C')|}$ the percentage of documents of category $C'$ in which $w$ occurs, $CF_C(w)$ the number of subcategories $C'' \in DirectSubCategories(C)$ such that $w$ occurs in a document $d \in Training(C'')$. Then the following measure: $v_i = TF_{c'}(w_i) \times DF^2_{c'}(w_i) \times \frac{1}{CF_c(w_i)}$ is used to select relevant tokens for the representation of documents in $C$.

Tokens that maximize $v_i$ ($maxTF \times DF^2 \times ICF$ criterion) are those commonly used in documents of category $C'$ but not in its sibling categories. The *category dictionary* of $C'$, $Dict_{C'}$, is the set of the best $n_{dict}$ terms with respect to $v_i$, where $n_{dict}$ is a user defined parameter.

For each learning task, $FeatSet_C = \bigcup_{C' \in DirectSubCategories(C)} Dict_{C'}$ represents the feature set and documents are represented according to the classical $TF \times idf$ measure [16].

### 3.2 Clustering-Based Relevant Example Selection

This approach follows the main idea of cluster sampling where the goal is to sample a set $S$ documents into $n_s$ subsets $N_1, N_2, \ldots, N_{n_s}$ respectively. These subsets (called strata) are non-overlapping, and together they comprise the whole of the data set (i.e., $\cup_{i=1..n_s} N_i = N$). When the strata have been determined, a sample is drawn from each stratum. Drawings are performed independently in different strata. Cluster sampling is often used in some applications where we wish to divide a heterogeneous data set into subsets, each of which is internally homogeneous [11]. For relevant examples selection, we consider the simple *k-means* [12] clustering algorithm for the identification of strata.

Once the strata have been identified, each cluster is represented by means of its surrogate. In our approach, as in the case of the Rocchio classifier [16], the surrogate of the cluster is its centroid $d'(i) = \sum_{d_j \in N_i} d_j / |N_i|$.

We also evaluate the opportunity of considering, in alternative to the centroid a representative example, that is, the example in $N_i$ that appears to be closer to the cluster centroid. Formally $d''(i) = \arg\min_{d_j \in N_i} d_1(d_j, d'(i))$, where $d_1(\cdot,\cdot)$ is the euclidean distance measure between document vectors.

For example reduction purposes, for each internal category $C$ of the hierarchy, both documents in $TS$ and in $WS$ are represented according to features in $FeatSet_C$ and according to the $TF \times idf$ measure.

### 3.3 Class Border identification for Relevant Example Selection

In this alternative approach to example reduction, the main idea is that of exploiting support vectors extracted by support vector machines [18] in order to identify the class border.

Indeed, in this case, the class is associated to the learning task that permits to establish whether an example should be passed down from a category $C$ to its descendant category $C'$ or not. This means that we extract a set of relevant examples by considering as positive examples the documents that belong to $Training(C')$ and as negative examples the documents that belong to $Training(C) - Training(C')$.

As in the case of clustering-based relevant example selection, this approach permits to reduce examples both in $TS$ and in $WS$.

Let $(\{\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_N, y_N)\}$ be the set of training documents in $Training(C)$ such that $\boldsymbol{x}_i \in \mathbb{R}^{|FeatSet_C|}$ ($\boldsymbol{x}_i$ is a document vector) and $y_i = +1$

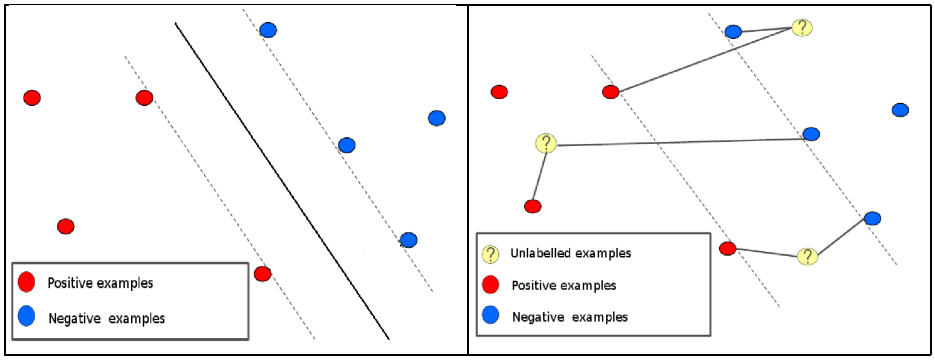


**Fig. 1.** a) Support vectors (examples on the dashed lines). b) Relevant examples selection from $WS$.

if $\boldsymbol{x}_i \in Training(C')$ and $y_i = -1$ if $\boldsymbol{x}_i \in Training(C) - Training(C')$. An SVM identifies the hyperplane in $\mathbb{R}^{|FeatSet_C|}$ that separates positive and negative examples with the maximum margin (*optimal separating hyperplane*). In general, the hyperplane can be constructed as the combination of all training examples, however, only some examples, called *support vectors*, do actually contribute to the optimal separating hyperplane which can be represented as:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i \boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}) + b \tag{1}$$

where $\boldsymbol{\Phi} : \mathbb{R}^{|FeatSet_C|} \rightarrow F$ is a non-linear map from $\mathbb{R}$ to another *feature space* $F$. Although the linear separability appears to be a strong limitation, as experimentally observed by [8], most text categorization problems are linearly separable.

Indeed, we are only interested in identifying support vectors, that is, vectors for which $\alpha_i \neq 0$ (see Figure 1a). The coefficients of the linear combination $\alpha_i$ and $b$ are determined by solving a large-scale quadratic programming (QP) problem, for which efficient algorithms that find the global optimum exist.

The SVM we use in the identification of support vectors is a modified version of the Sequential Minimal Optimization classifier (SMO) [13] with linear kernels. SMO is very fast and is based on the idea of breaking a large QP problem down into a series of smaller QP problems that can be solved analytically. This allows us to directly identifying document vectors $\boldsymbol{x}_i$ for which $\alpha_i \neq 0$.

Let $D(C, C') = \{\boldsymbol{x}_i | x_i \in Training(C), \alpha_i\}$ be the set of support vectors that have been identified, they are used in order to identify the subset of documents in $WS$ to be considered in the classification phase. In particular, we are only interested in keeping the most discriminative $p\%$ documents from $WS$ according to the score function $score : WS \rightarrow \mathbb{R}$ defined as follows:

$$score(\boldsymbol{d}) = \min_{\boldsymbol{x}_i \in Training(C');\ \boldsymbol{x}_j \in Training(C) - Training(C')} d_1(\boldsymbol{d}, \boldsymbol{x_i}) + d_1(\boldsymbol{d}, \boldsymbol{x_j}) \tag{2}$$

Intuitively, we only consider examples in $WS$ that are close to both positive and negative class margins.

## 4 SGT Hierarchical Classifier

In this section, we first introduce the hierarchical transductive classification, and, then, we detail the application of SGT.

### 4.1 Hierarchical Transductive Classification

We assume that a classifier returns a numerical score $\gamma_{C \to C'}(d)$ that expresses a "belief" that a document $d$ belonging to $C$ also belongs to a direct subcategory $C'$. The document $d$ is passed down if $\gamma_{C \to C'}(d)$ is greater than a threshold, which is automatically determined for each class by an algorithm that minimizes, on the training set, a tree distance. This distance measures the number of edges in the hierarchy of categories between the actual class of a document and the class returned by the hierarchical classifier [3].

As in [3], the computation proceeds bottom-up, from leaves to the root. The difference is that in this work we learn, for each internal category $C$, $m$ two-class classifiers, one for each subcategory $C'$ and compare the scores. This is quite different from what proposed in [3], where a 1-of-m classifier is learnt for each internal node. This would permit us to exploit two-class classifiers and avoid computational problems coming from a pair-wise coupling classification [7].

The classifier used to classify examples belonging to internal nodes of the hierarchy is based on the Spectral Graph Transducer algorithm (SGT) proposed in [9] that works in the transductive setting. Although, in its final formulation SGT returns hard class assignments, we use the SGT algorithm in order to compute the scores $\gamma_{C \to C'}(d)$. This way, the algorithm can be used both to compute thresholds and to classify examples in the working set. The problem solved by each application of SGT can be formalized as follows:

Given: An internal category $C$; A direct subcategory $C'$ of $C$; A set of $l$ relevant labeled examples (documents) belonging to $C$ and its descendants (identified as specified in Section 3). Positive examples (+1) refer to documents in $Training(C')$ and all its descendants, while negative examples (-1) refer to all other examples in categories descendants of $C$ (in $Training(C) - Training(C')$); A set of relevant unlabeled examples (possibly) belonging to $C$ and its descendants. The task of the transductive algorithm is to compute the score $\gamma_{C \to C'}(d)$ for each relevant document $d$ in the training or in the working set.

### 4.2 Application of SGT algorithm

The algorithm builds a nearest neighbor graph $G = (N, E)$, with labeled and unlabeled examples as vertexes, and dissimilarity measure $(d_2(d_i, d_j))$ between the neighboring examples as edge weights. SGT assigns labels to unlabeled examples by cutting G into two subgraphs $G^-$ and $G^+$, and tags all examples

corresponding to vertexes in $G^-$ ($G^+$ ) with -1 (+1). To give a good prediction of labels for unlabeled examples, SGT chooses the cut of G that maximizes the normalized cut cost: $\max_y \frac{cut(G^+,G^-)}{|\{i|y_i=+1\}||\{i|y_i=-1\}|}$ where $y = [y_i]_{\{i=1,..,n\}}$ is the prediction vector (where $n$ is the number of both labeled and unlabeled relevant examples), and $cut(G^+, G^-)$ is the sum of the weights of all edges that cross the cut (i.e., edges with one end in $G^-$ and the other in $G^+$). The optimization is subjected to the following constraints: *(i)* $y_i \in \{-1, +1\}$ and *(ii)* labels for labeled training examples must be correct, i.e., vertexes corresponding to positive (negative) labeled relevant training examples must lie in $G^+$ ($G-$). As this optimization is NP-hard, SGT performs approximate optimization by means of a spectral graph method which solves the following problem [5]:

$$\min_Z \quad Z^T L Z + c(Z - y)^T C(Z - y) \tag{3}$$

such that $Z^T 1 = 0$ and $Z^T Z = n$ and where

- $Z$ is the transformed prediction vector with comparable scores,
- $L$ is computed as the Laplacian matrix $L = (B - A)$ in the case of RATIO CUT or, alternatively, as the normalized Laplacian matrix obtained as $L = B^{-1}(B - A)$ in the case of NORMALIZED CUT [17];
- $A = [a_{i,j}]_{\{i,j=1,..,n\}} = [a'_{i,j} + a'_{j,i}]_{\{i,j=1,..,n\}}$ where $a'_{i,j} = d_2(d_i, d_j)$;
- $B = [b_{i,j}]_{\{i,j=1,..,n\}}$ is the diagonal matrix such that $b_{i,i} = \sum_j a_{i,j}$;
- $c$ is a user-defined parameter;
- $C = [c_{i,j}]_{\{i,j=1,..,n\}}$ is a diagonal cost matrix with $c_{i,i} = l/(2l+)$ for positive relevant examples, $c_{i,i} = l/(2l-)$ for negative and $c_{i,i} = 0$ for unlabelled relevant examples;
- $l+$ ($l-$)is the number of positive (negative) relevant labeled examples and $l \leq n$ is the number of relevant labelled examples;
- $\gamma = [\gamma_i]_{\{i=1,..,n\}}$ is a vector with $\gamma_i = \sqrt{l - /l+}$ for positive examples, $\gamma_i = \sqrt{l + /l-}$ for negative examples and $\gamma_i = 0$ for unlabelled examples.

This minimization problem leads to compute $Z^* = V(M - \lambda^* I)^{-1} b$ where $V$ is the matrix with all eigenvectors of $L$ except the smaller; $b = CV^T C\gamma$; $M = (D + cV^T I)$; $D$ is the diagonal matrix with the square of all eigenvalues of $L$ except the smaller; $\lambda^*$ is the smaller eigenvalue of $\begin{bmatrix} M & -I \\ \frac{-1}{n} bb^T & M \end{bmatrix}$.

The vector $Z^* = [z_i^*]_{\{i=1,..,n\}}$ is then used to compute the score $\gamma_{C \to C'}(d_i)$:

$$\gamma_{C \to C'}(d_i) = (z_i^* - \min_j z_j^*)/(\max_j z_j^* - \min_j z_j^*) \tag{4}$$

The used dissimilarity measure $d_2(\cdot, \cdot)$ is the cosine dissimilarity $d_2(d_i, d_j) = 1 - \frac{\boldsymbol{d_i} \cdot \boldsymbol{d_j}}{\|\boldsymbol{d_i}\|_2 \|\boldsymbol{d_j}\|_2}$, where $\boldsymbol{d_i}$ ($\boldsymbol{d_j}$) represents the $TF \times idf$ representation of $d_i$ ($d_j$).

## 5 Experiments

To evaluate the applicability of the proposed approach, we performed experiments on distinct experimental settings involving two distinct datasets. As base-

line we considered the Hierarchical SGT transductive classifier that do not exploit relevant example selection [2].

Results are obtained with the following parameters: $c = 10^4$ as proposed in [9]; $n_{dict}$=100; $\alpha = 0.4$; $K$ used in the K-NN label propagation is set to the highest odd integer such that $K \leq \sqrt{n}$ (according to [19]) where $n$ is the total number of both labelled and unlabelled examples that (possibly) belong to the processed category $C$; $n_s = 5\% \times n$; $p\% = 5\%$. Values of $n_{dict}$ and $\alpha$ are estimated after an empirical evaluation. Values of $n_s$ and $p\%$ are set in order to make the comparison between relevant example selection algorithms fair.

Results obtained with the different experimental settings aim at comparing the trasductive algorithm without example reduction (HSGT- Hierarchical SGT) with the trasductive algorithm with relevant example selection based on k-means clustering and $d'(\cdot)$ cluster representation (KmSGT), the trasductive algorithm with relevant example selection based on k-means clustering and $d''(\cdot)$ cluster representation (SelectSGT) and the trasductive algorithm with relevant example selection based on support vectors (SVSGT).

### 5.1 Datasets

**Reuters Corpus Volume I (RCV1)** RCV1 is a benchmark dataset widely used in text categorization and in document retrieval[1] consisting of over 800,000 newswire stories organized in a set of 104 categories distributed on 4-levels.

We pre-processed documents as proposed by Lewis et al. and, in addition, we considered only documents associated to a single category. This selection is due to the fact that in this study we are interested in investigating single category assignment [3]. We separated the training set and the testing set using the same split adopted by Lewis et al. In particular, documents published from August 20, 1996 to August 31, 1996 were included in the training set, while documents published from September 1, 1996 to August 19, 1997 were included in the working set. The result was a split of the 804,414 documents into 23,149 training documents and 781,265 working documents. After multiple-label document removal, we had 150,765 documents, (4,517 training documents and 146,248 testing documents). In our experiments we analyze three large subsets of RCV1: A subset rooted in the category "*C3*" (1,647 training documents, 50,345 working documents); a subset rooted in the category "*C18*" (1,438 training documents, 44,148 working documents); a subset rooted in the category "*MCAT*" (10,715 training documents, 163,592 working documents).

**Dmoz dataset** Dmoz data is obtained from the documents referenced by the Open Directory Project (ODP) [2]. We extracted all documents referenced at the top five levels of the directory rooted in "*Health\Conditions_and_Diseases*". Documents containing only scripts and documents whose size is less than 3Kb are removed. At the end, the dataset contains 3,668 documents in 203 categories.

[1] The dataset cannot be made available on-line without maintainers authorization.

[2] The dataset is available at http://www.di.uniba.it/%7ececi/micFiles/dmoz_health_conditions_and_diseases_docs.zip.

| DATASET | cut | HSGT | KmSGT | SelectSGT | SVSGT |
|---|---|---|---|---|---|
| CANCER | RATIO | 64% | 62% | 55% | 29% |
| | NORMALIZED | 60% | 56% | 53% | 32% |
| CARDIOVASCULAR | RATIO | 63% | 61% | 53% | 31% |
| | NORMALIZED | 60% | 55% | 47% | 42% |
| CONDITION | RATIO | 37% | 29% | 26% | 12% |
| | NORMALIZED | 34% | 23% | 26% | 12% |
| C3 | RATIO | – | 30% | 32% | 32% |
| | NORMALIZED | – | 29% | 7% | 29% |
| C18 | RATIO | – | 65% | 68% | 49% |
| MCAT | RATIO | – | 49% | 50% | %5 |

**Table 1.** Average accuracies obtained with HSGT, KmSGT, SelectSGT and SVSGT. Thresholds are obtained on the whole set of training examples.

The dataset is analyzed by means of a 3-fold cross-validation (CONDITION). Two subset of this dataset rooted in the category "*Cancer*" and in the category "*Cardiovascular disorders*" respectively are also analyzed by means of a 3-fold cross-validation. It is noteworthy that, differently from usual, in this paper the $t$-fold cross-validation uses in turn one fold for training and the remaining $t-1$ folds as working set. This is coherent with principles motivating the transductive approach where the working set is generally larger than the training set.

### 5.2 Results

Accuracy results[3] are reported in Tables 1 and 2. In particular, Table 1 shows that relevant example selection permits to obtain classification accuracies that are lower (but similar) than those obtained with *HSGT*. By analyzing Table 2 it is possible to see that when considering only relevant examples in automatic threshold determination, accuracy significantly increases. In fact, in most of cases *KmSGT* outperforms *HSGT* even if it works on smaller set of examples. A possible reason can be found in the fact that, in this way, the classification performed by SGT is coherent with the automatic threshold determination phase.

By comparing results obtained with relevant example selection, we can see that the clustering algorithm permits to identify a good representative set of examples to be used during the learning phase. We cannot draw the same conclusion for *SVSGT* that, as SMO [13], suffers from the high imbalanced distribution of examples. In fact, for C3 and C18, where categories are almost uniformly distributed *SVSGT* provides interesting results. It is also noteworthy that the RATIO CUT outperforms the NORMALIZED CUT both in terms of accuracy and efficiency (for this reason we do not report NORMALIZED CUT results for C18 and MCAT). This means that the use of a normalized cut in transductive learning is not as beneficial as in the case of image processing [17].

Finally, results reported in Table 3 give a clear perspective of the learning time reduction obtained with relevant example selection. This advantage is more clear when thresholds are determined only on relevant examples.

[3] Due to space complexity problems, it was not possible to run HSGT on the datasets C3, C18 and MCAT.

| DATASET | cut | HSGT | KmSGT | SelectSGT | SVSGT |
|---|---|---|---|---|---|
| CANCER | RATIO | 64% | 65% | 59% | 32% |
| | NORMALIZED | 60% | 60% | 54% | 28% |
| CARDIOVASCULAR | RATIO | 63% | 64% | 57% | 33% |
| | NORMALIZED | 60% | 60% | 51% | 34% |
| CONDITION | RATIO | 37% | 39% | 36% | 13% |
| | NORMALIZED | 34% | 37% | 32% | 14% |
| C3 | RATIO | – | 31% | 28% | 18% |
| | NORMALIZED | – | 31% | 12% | 29% |
| C18 | RATIO | – | 69% | 77% | 70% |
| MCAT | RATIO | – | 49% | 50% | 5% |

**Table 2.** Average accuracies obtained with HSGT, KmSGT, SelectSGT and SVSGT. Thresholds are obtained on the set of relevant training examples.

| DATASET | HSGT | KmSGT | | SelectSGT | | SVSGT | |
|---|---|---|---|---|---|---|---|
| | | NS | S | NS | S | NS | S |
| CANCER | 64 | 56 | 50 | 53 | 49 | 45 | 42 |
| CARDIOVASCULAR | 63 | 51 | 43 | 41 | 40 | 41 | 42 |
| CONDITION | 58803 | 19939 | 10482 | 16200 | 6971 | 11685 | 7800 |
| C3 | – | 35991 | 16541 | 12581 | 16881 | 12354 | 30762 |
| C18 | – | 25801 | 35000 | 15440 | 16671 | 4853 | 3203 |
| MCAT | – | 413548 | 253410 | 168962 | 175239 | 234431 | 62072 |

**Table 3.** (Average) classification times with Ratio cut (in secs.). *NS* refers to thresholds obtained on the whole set of training examples. *S* refers to thresholds obtained on the set of relevant training examples.

## 6 Conclusions

In this paper, we present a novel approach for automatic classification of documents into a hierarchy of categories that exploits relevant example selection and works in the transductive setting. The proposed approach is based on a framework that exploits the SGT classifier in internal nodes of the hierarchy. This way, it can pass down examples to more specific categories on the basis of scores returned by the classifier. Documents can also be classified in internal nodes of the hierarchy according to some automatically learned thresholds. The SGT algorithm is used both for learning thresholds and for classifying examples.

Relevant example selection is performed according to two different approaches: the first approach reduces the number of documents by exploiting clustering algorithms, while the second approach identifies and keeps only documents that are at the boundary of the class. Results empirically prove that relevant example selection based on clustering algorithms permits to tame the computational complexity and, at the same time, permits to increase predictive capabilities.

For future work, we intend to *i*) extend experiments in order to give more insight about parameter tuning and *ii*) exploit the proposed approach in multi-label classification where classification takes into account multiple dimensions.

## Acknowledgment

This work is partial fulfillment of the research objective of the project “DM19410 - Laboratorio di Bioinformatica per la Biodiversità a Molecolare”.

## References


1. A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1–2):245–271, 1997.
2. M. Ceci. Hierarchical text categorization in a transductive setting. In *ICDM Workshops*, pages 184–191. IEEE Computer Society, 2008.
3. M. Ceci and D. Malerba. Classifying web documents in a hierarchy of categories: a comprehensive study. *J. Intell. Inf. Syst.*, 28(1):37–78, 2007.
4. Y. Chen, G. Wang, and S. Dong. Learning with progressive transductive support vector machines. *Pattern Recognition Letters*, 24:1845–1855, 2003.
5. I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD '01*, pages 269–274, New York, NY, USA, 2001. ACM.
6. S. Dumais and H. Chen. Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 256–263. ACM Press, 2000.
7. T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *NIPS '97*, pages 507–513. MIT Press, 1998.
8. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 137–142. Springer-Verlag, 1998.
9. T. Joachims. Transductive learning via spectral graph partitioning. In *ICML 2003*. Morgan Kaufmann, 2003.
10. M. Kukar and I. Kononenko. Reliable classifications with machine learning. In *Proc. of the 13th European Conference on Machine Learning, ECML 2002*, pages 219–231. Springer-V., 2002.
11. H. Liu and H. Motoda. On issues of instance selection. *Data Min. Knowl. Discov.*, 6(2):115–130, 2002.
12. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
13. J. Platt. *Advances in kernel methods - support vector learning*, chapter Fast training of support vector machines using sequential minimal optimization. 1998.
14. M. F. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, pages 313–316, 1997.
15. M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Inf. Retr.*, 5(1):87–118, 2002.
16. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
17. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.
18. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
19. D. Wettschereck. *A study of Distance-Based Machine Learning Algorithms*. PhD thesis, Oregon State University., 1994.
20. D. R. Wilson and T. R. Martinez. Instance pruning techniques. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 403–411, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
21. Y. Yang. Sampling strategies and learning efficiency in text categorization. In *In AAAI Spring Symposium on Machine Learning in Information Access*, pages 88–95, 1996.