Hierarchical Text Categorization in a Transductive Setting

Michelangelo Ceci
University of Bari - Dipartimento di Informatica
Via Orabona, 4 - 70125 Bari - Italy
ceci@di.uniba.it

Abstract

Transductive learning is the learning setting that permits to learn from "particular to particular" and to consider both labelled and unlabelled examples when taking classification decisions. In this paper, we investigate the use of transductive learning in the context of hierarchical text categorization. At this aim, we exploit a modified version of an inductive hierarchical learning framework that permits to classify documents in internal and leaf nodes of a hierarchy of categories. Experimental results on real world datasets are reported.

1 Introduction

Recently, there has been a growing interest in learning algorithms capable of utilizing both labeled and unlabeled data for prediction tasks, such as classification. The reason for this attention is the high cost of manually assigning labels and the high availability of unlabelled examples. Two main settings have been proposed in the literature to exploit information contained in both labeled and unlabeled data: the *semi-supervised setting* and the *transductive setting* ([21]). The former is a type of inductive learning, since the learned function is used to make predictions on any possible example. The latter asks for less – it is only interested in making predictions for the given set of unlabeled data. Since transduction needs no general hypothesis, it appears to be an easier problem than (semi-supervised) induction and it is likely to become much more popular in the future. Several transductive learning methods have been proposed in the literature for SVMs ([2] [8] [10] [4]), for k-NN classifiers ([11]) and even for general classifiers ([12]).

In this paper we propose a hierarchical text classifier that works in the transductive setting. Hierarchical text classification is the process of automatically assigning one or more predefined categories to text documents [13, 15, 6, 16, 18, 23], where the pre-defined categories are organized in a tree-like structure. 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. Yahoo, Google Directory, Medical Subject Headings (MeSH), Open Directory Project and Reuters Corpus Volume I provides typical examples of organization of documents in topic hierarchies.

The structural relationship among categories can be taken into account when devising the classification process. While in flat classification a given document 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 documents among categories in a top-down way. The advantage of this hierarchical view of the classification process is that the problem is partitioned into smaller subproblems, each of which can be effectively and efficiently managed. Another motivation comes from the observation that both precision and recall decrease as the number of categories increases [1, 25], due to the increasing effect of term polysemy for large corpora.

Although, a wide range of supervised learning algorithms have been proposed in the literature for hierarchical classification in the inductive learning setting [3], to our our knowledge no study has been made on hierarchical classification in the transductive setting. This paper contributes to fill this gap by investigating the application of a flat transductive classifier, namely the Spectral Graph Transducer (SGT) [11], in the context of a hierarchical classification framework. Scalability issues due to the high computational cost of the trasductive classifier are mitigated by reducing the number of training examples through a distance based method which removes those labeled examples that are far from a class boundary.

This paper is organized as follows. The problem to solve and the background work are introduced in the next section. The proposed solution is described in Section 3 and the example reduction algorithm is presented in Section 4. Experimental results on real world datasets are reported in Section 5 while conclusions are drawn in Section 6.

978-0-7695-3503-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDMW.2008.126

2 Problem Definition and Background work

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:

- 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)
- a working set WS of unlabelled documents;

Find: a prediction of the class value of each document in the working set WS which is as accurate as possible.

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 training set associated to each category may or may not include training documents of subcategories. Fourth, the classifier may or may not take into account the hierarchical relation between categories. Fifth, a stopping criterion is required for hierarchical classification of new documents in non-leaf categories. Sixth, performance evaluation criteria should take into account the hierarchy when considering classification errors.

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 a system, named WebClass, which is briefly introduced in the following.

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*. An example is illustrated in Figure 1.

During the classification process, the 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. These different representations of a document make the classification scores incomparable across different nodes in the hierarchy and prevent the correct application of an exhaustive search strategy instead of the proposed greedy strategy.

During the training process, a classifier is learned for each internal category c of the hierarchy. This classifier is used to decide, during the classification of a new document, which category c' among the direct subcategories of c is the most appropriate to receive the document. In general, however, a document should not be necessarily passed down to a subcategory of c. This makes sense in the case that:

1. the document to be classified deals with a general rather than a specific topic, or
2. 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 $\gamma_{C \rightarrow C'}(d)$ 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 temporary assigned to a category C will be passed down to a category C' if $\gamma_{C \rightarrow C'}(d) > Th_C(C')$, 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 tries to minimize a measure based on a *tree distance*[3].

3 Using SGT in hierarchical classification

In this section, the proposed method is described in detail. In particular, we first explain how documents are represented at different levels of the hierarchy, then we introduce the hierarchical transductive classification, and, finally, we detail the application of SGT.

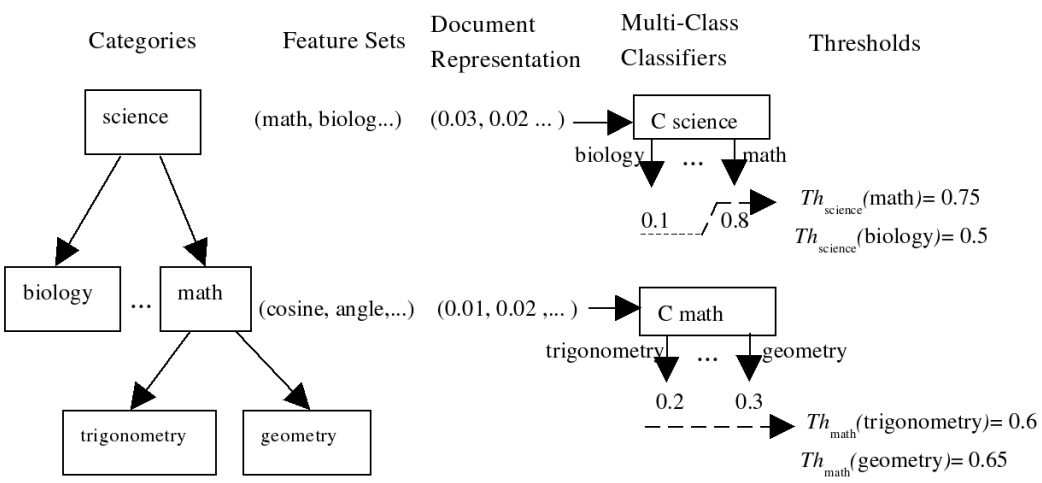

Figure 1. Classification of a new document. On the basis of the scores returned by the first classifier (associated to the category *science*) the document is passed down to *math*. The scores returned by the second classifier (associated to the category *math*), are not high enough to pass down the document to either *trigonometry* or *geometry*. Therefore, the document is classified in the *math* category.

3.1 Document preprocessing and representation

Document preprocessing is necessary in order to

1. Remove *stopwords*, such as articles, adverbs, prepositions and other frequent words.

2. Determine equivalent stems (*stemming*) by means of Porter's algorithm for English texts [17].

Training documents are subsequently 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,
• $Training(C)$ 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:

$$v_i = TF_{c'}(w_i) \times DF^2_{c'}(w_i) \times \frac{1}{CF_c(w_i)} \qquad (1)$$

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, the following feature set is used: $FeatSet_C = \bigcup_{C' \in DirectSubCategories(C)} Dict_{C'}$ and documents are represented according to the classical $TF \times idf$ measure [20].

3.2 Hierarchical Transductive Classification

When working in the transductive setting, we do not distinguish between learning and classification steps. However, the hierarchical organization of categories requires the 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.

In this work 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 [7].

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 subsequent pair-wise coupling classification [9].

The classifier used to classify examples belonging to internal nodes of the hierarchy is based on the Spectral Graph Transducer algorithm (SGT) proposed in [11] 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 labeled examples (documents) belonging to C and its descendants. Positive examples (labeled with +1) refer to documents in $Training(C')$ and all its descendands, while negative examples (labeled with -1) refer to all other examples in categories descendants of C (in $Training(C) - Training(C')$);
- A set of 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 document d in the training or in the working set (labelled or unlabelled) such that error is minimized.

Thresholds are computed by considering only the training set (excluding unlabelled examples), while classification is performed by considering both training and working set.

3.3 Application of SGT algorithm

The algorithm builds a nearest neighbor graph $G = (N, E)$, with labeled and unlabeled examples as vertexes, and dissimilarity measure ($d(d_i, d_j)$) 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.

$$\max_y \frac{cut(G^+, G^-)}{|\{i|y_i = +1\}||\{i|y_i = -1\}|} \tag{2}$$

where $y = [y_i]_{\{i=1,..,n\}}$ is the prediction vector (where n is the number of both labeled and unlabeled 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 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]:

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

$$\text{such that } Z^T 1 = 0 \text{ and } Z^T Z = n$$

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 [22];
- $A = [a_{i,j}]_{\{i,j=1,..,n\}} = [a'_{i,j} + a'_{j,i}]_{\{i,j=1,..,n\}}$ where $a'_{i,j} = d(d_i, d_j)$;
- $B = [b_{i,j}]_{\{i,j=1,..,n\}}$ 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 examples, $c_{i,i} = l/(2l-)$ for negative and $c_{i,i} = 0$ for unlabelled examples;
- $l+$ ($l-$)is the number of positive (negative) labeled examples and $l \leq n$ is the number of 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

$$Z^* = V(M - \lambda^* I)^{-1} b \tag{4}$$

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; λ^* 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 $\gamma_{C \to C'}(d_i)$. In particular:

$$\gamma_{C \to C'}(d_i) = z_i^* - \min_j z_j^* \tag{5}$$

The dissimilarity measure $d(\cdot, \cdot)$ used in this work is the cosine dissimilarity computed as follows:

$$d_1(d_i, d_j) = 1 - \frac{\vec{d_i} \cdot \vec{d_j}}{\left\|\vec{d_i}\right\|_2 \left\|\vec{d_j}\right\|_2} \tag{6}$$

where $\vec{d_i}$ ($\vec{d_j}$) represents the $TF \times idf$ representation of d_i (d_j). This normalization represents the cosine of the angle spanned by the two vectors $\vec{d_i}$ and $\vec{d_j}$. It is a dissimilarity measure, therefore, the lower the value, the more similar the documents d_i and d_j.

4 Distance-Based Training Set Reduction

Differently from the classical inductive setting, in the transductive setting both labeled and unlabeled documents are used during learning. This poses additional computational problems mainly due to the demanding storage required to process documents. The problem is due to the the size of matrices in Equation (4), where, even the sparse matrix representation we use, does not help to solve the problems (it is necessary to compute inverses).

In order to limit this problem, in this work, an example reduction algorithm is used. The proposed solution works on labeled documents and exploits the graph G introduced in the previous section.

As the transductive algorithm is a distance based algorithm, the example reduction algorithm follows the same approach and removes labeled examples that are somehow distant from the class boundary. The algorithm is based on the definition of chain of examples defined as follows:

Definition [Chain of examples] Given a graph $G = (N, E)$, given a node (example) $d_0 \in N$ and a dissimilarity measure $d(\cdot, \cdot)$. Then a chain of examples $ce = (d_0, d_1, d_2, ...d_s)$ (where $s \leq l$ is not defined apriori) satisfies the following conditions:

- $label(d_{i+1}) \neq label(d_i) \qquad \forall i = 0, \ldots, s-1$
- $d(d_{i+1}, d_i) \leq d(d_i, d_{i-1}) \qquad \forall i = 1, \ldots, s-1$
- $d_i \neq d_j \quad \forall i, j = 0, \ldots, s \quad i \neq j$

The algorithm (see Algorithm 1) works iteratively and, at each iteration, it chooses a seed labelled example and, from that example it identifies a chain. All examples that are sufficiently distant (see line 8) from examples of the opposite class are removed. The algorithm stops the search when all examples have been considered at least once in a chain. α is a user-defined parameter that allows the user to specify the algorithm selectivity.

Algorithm 1 Example reduction

1: **reduce_examples**(graph $G(N, E)$)
2: $N' \leftarrow \emptyset$;
3: $N_r \leftarrow \emptyset$;
4: **while** $N/N' \neq \emptyset$ **do**
5: let $d_0 \in N/N'$;
6: $C = (d_0, \ldots d_p) \leftarrow$ build_chain(d_0,N);
7: $N' \leftarrow N' \cup C$;
8: $N_r \leftarrow N_r \cup \{d_i \in C | \alpha \cdot d(d_i, d_{i+1}) > d(d_{p-1}, d_p)\}$;
9: **end while**
10: **return** N/N_r

The algorithm is applied before each learning task.

5 Experiments

To evaluate the applicability of the proposed approach, we performed experiments on four distinct experimental settings involving two distinct datasets. As baseline we considered the inductive counterpart of the proposed algorithm, that is, a hierarchical version of the original K-NN algorithm ([14]) that exploits the inductive WebClass framework. Here, K is the highest odd integer such that $K \leq \sqrt{n}$ (according to [24]) and the score function is defined as:

$$\gamma_{C \to C'}(d) = \frac{|\{d_i \in N_K(d) | d_i \in Training(C')\}|}{K}, \tag{7}$$

where $N_K(d)$ is the set of labelled examples in the k-neighborhood of d according to the distance function $d(\cdot, \cdot)$.

Results are obtained with the following parameters: $c = 10^4$ as proposed in [11]; n_{dict}=100 and $\alpha = 0.4$. Values of n_{dict} and α are estimated after an empirical evaluation.

Results obtained with the four experimental settings aim at comparing the trasductive algorithm without example reduction (that we indicate in these experiments as HSGT-Hierarchical SGT), the trasductive algorithm with example reduction (HeSGT - Hierarchical example reduction SGT) with its inductive counterpart (HK-NN - Hierarchical K-NN classifier). In addition, results also aim at comparing HSTG with HeSGT in terms of accuracy, time reduction and training set reduction.

5.1 Datasets

5.1.1 Dmoz dataset

The first data set used in this experimental study is obtained from the documents referenced by the Open Directory Project (ODP) (www.dmoz.org)[1]. We extracted all actual Web documents referenced at the top five levels of the Web directory rooted in the branch "*Health\Conditions_and_Diseases*". Empty documents, documents containing only scripts, and documents whose size is less than 3Kb are removed. At the end, the dataset contains 3,668 documents organized in 203 categories.

The dataset is analyzed by means of a 3-fold cross-validation (CONDITION_3). A subset of this dataset rooted in the category "*Cancer*" is also analyzed by means of a 3-fold cross-validation (CANCER_3) and a 10-fold cross-validation (CANCER_10). 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.

5.1.2 OHSUMED dataset

The OHSUMED test collection is a set of 348,566 references from MEDLINE, the on-line medical information database, consisting of titles and/or abstracts from 270 medical journals over a five-year period (1987-1991). ¿From the original OHSUMED test collection, we pre-processed documents in order to remove those without abstract and empty categories. In addition, we considered only documents associated to a single category. This selection is due to the fact that in this study we have investigated single category assignment (the proposed method is based on the assumption that a document can be assigned to one category at the most). The removal of documents associated with multiple classes has also been adopted by other authors on different datasets in the evaluation of single-label corpora [19].

¿From the pre-processed dataset we considered the subtree rooted in the category "*Substance related disorders*" containing 1054 documents in 10 categories organized in a tree-levels hierarchical structure. This dataset has been analyzed according to a 3-fold cross validation (SUBSTANCE_3).

5.2 Results

Accuracy results are reported in Table 1. These results show that the transductive approach, in most of cases, significantly improves its inductive counterpart. The difference becomes evident in the case of a 10-fold cross-validation. This confirms that the transductive approach takes great advantage from unlabelled examples. An exception is represented by SUBSTANCE_3 setting, where highly unbalanced distribution of examples in the hierarchy of categories makes accuracy results of HK-NN comparable with HSGT and HeSGT. It is also noteworthy that the RATIO CUT outperforms the NORMALIZED CUT both in terms of accuracy and efficiency. This means that the use of a normalized cut in transductive learning is not as beneficial as in the case of image processing [22].

Different statistics are reported in Table 2 where reductions in learning time, training set size and accuracy of HeSGT with respect to HSGT are reported. Among the different datasets, by keeping unchanged the t value in the t fold cross-validation the percentage of removed training examples remains unchanged. As expected, the situation is different for the 10-fold cross-validation, where the smaller size of the training set does not justify a high number of removed examples. By balancing learning time reduction and accuracy reduction we can conclude that in most of cases it is convenient to exploit the algorithm that considers the example reduction (HeSGT), especially in the case of RATIO CUT. An exception is represented by the setting with the largest dataset (CONDITION_3). However, in this case, results reported in the table are due to generally low values of accuracy. In fact, the accuracy difference is comparable to other experimental settings (see Table 1).

In Figure 2, the distribution of errors is shown. In particular, misclassification error (percentage of documents misclassified into a category not related to the correct category in the hierarchy), non-classification error (percentage of documents classified in the root), generalization error (percentage of documents misclassified into a supercategory of the correct category) and specialization error (percentage of documents misclassified into a subcategory of the correct one) are reported. Generally, the type of committed errors changes from one setting to another. However, except for SUBSTANCE_3, a significant portion of errors is due to "less serious" errors (generalization + specialization errors).

6 Conclusions

In this paper, we present a novel approach for automatic classification of documents into a hierarchy of categories that 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 unla-

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

		Transductive		Inductive
DATASET	**Type of cut**	**HSGT**	**HeSGT**	**HK-NN**
CANCER_3	RATIO	64%	61%	38%
	NORMALIZED	60%	54%	
CANCER_10	RATIO	56%	53%	7%
	NORMALIZED	47%	42%	
CONDITION_3	RATIO	37%	26%	10%
	NORMALIZED	33%	25%	
SUBSTANCE_3	RATIO	70%	73%	72%
	NORMALIZED	73%	68%	

Table 1. Average accuracies obtained with HSGT, HeSGT and HK-NN

DATASET	Type of cut	Learning time reduction %	Training Set Reduction%	Accuracy Reduction%
CANCER_3	RATIO	21%	67%	5%
	NORMALIZED	26%	67%	10%
CANCER_10	RATIO	8%	14%	5%
	NORMALIZED	-6%	14%	11%
CONDITION_3	RATIO	54%	61%	30%
	NORMALIZED	48%	61%	24%
SUBSTANCE_3	RATIO	36%	58%	-4%
	NORMALIZED	45%	58%	7%

Table 2. HeSGT vs HSGT

belled examples. To tame the computational complexity of the transductive approach, an example reduction algorithm for labelled examples is also integrated in the framework.

Results on real world datasets show significant advantages of the proposed approach over its inductive counterpart and show the effectiveness, in most of cases, of the example reduction algorithm. For future work, we will exploit clustering algorithms in order to also reduce working set during classification.

7 Acknowledgments

The work presented in this paper is partial fulfillment of the research objective set by the project D.A.M.A. “Document Acquisition, Management and Archiving”.

References

[1] C. Apté, F. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *Information Systems*, 12(3):233–251, 1994.

[2] K. P. Bennett. Combining support vector and mathematical programming methods for classification. pages 307–326, 1999.

[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 *KDD ’01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 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] F. Esposito, D. Malerba, V. Tamma, and H. Bock. *Classical resemblance measures*, volume 15 of *Studies in Classification, Data Analysis, and Knowledge Organization*, chapter Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data, pages 139–152. Springer-Verlag, 2000.

[8] A. Gammerman, K. Azoury, and V. Vapnik. Learning by transduction. In *Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence, UAI 1998*, pages 148–155. Morgan Kaufmann, 1998.

[9] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *NIPS ’97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 507–513. MIT Press, 1998.

[10] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the 16th International Conference on Machine Learning, ICML 1999*, pages 200–209. Morgan Kaufmann, 1999.

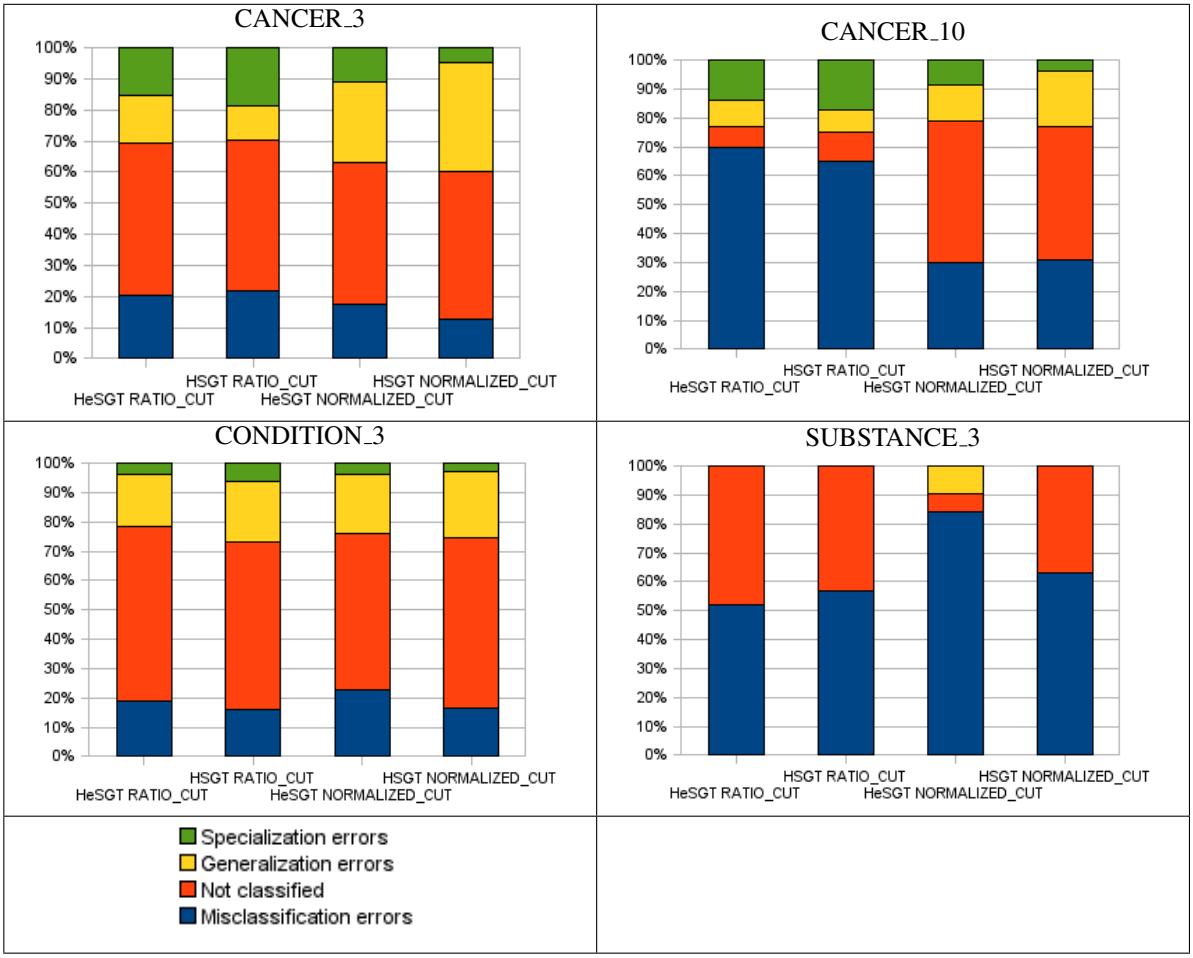

Figure 2. Distribution of errors

[11] T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. of the 20th International Conference on Machine Learning, ICML 2003*. Morgan Kaufmann, 2003.

[12] 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.

[13] A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367. Morgan Kaufmann Publishers Inc., 1998.

[14] T. Mitchell. *Machine Learning*. McGraw Hill, New York, USA, 1997.

[15] D. Mladenić. *Machine learning on non-homogeneus, distributed text data*. PhD thesis, University of Ljubjana, Ljubjana, Slovenia, 1998.

[16] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perception learning, and a usability case study for text categorization. *SIGIR Forum*, 31(SI):67–73, 1997.

[17] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[18] M. E. Ruiz and P. Srinivasan. Hierarchical text categorization using neural networks. *Inf. Retr.*, 5(1):87–118, 2002.

[19] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Mach. Learn.*, 39(2-3):135–168, 2000.

[20] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[21] M. Seeger. Learning with labeled and unlabeled data, 2000.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. volume 22, pages 888–905, Washington, DC, USA, 2000. IEEE Computer Society.

[23] A. S. Weigend, E. D. Wiener, and J. O. Pedersen. Exploiting hierarchy in text categorization. *Inf. Retr.*, 1(3):193–216, 1999.

[24] D. Wettschereck. *A study of Distance-Based Machine Learning Algorithms*. PhD thesis, Oregon State University., 1994.

[25] Y. Yang. An evaluation of statistical approaches to medline indexing. In *Proceedings of the AMIA*, pages 358–362, 1996.