Efficient Path Prediction for Semi-Supervised and Weakly Supervised Hierarchical Text Classification

Huiru Xiao
Hong Kong University of Science and Technology
hxiaoaf@cse.ust.hk

Xin Liu
Hong Kong University of Science and Technology
xliucr@cse.ust.hk

Yangqiu Song
Hong Kong University of Science and Technology
yqsong@cse.ust.hk

ABSTRACT

Hierarchical text classification has many real-world applications. However, labeling a large number of documents is costly. In practice, we can use semi-supervised learning or weakly supervised learning (e.g., dataless classification) to reduce the labeling cost. In this paper, we propose a path cost-sensitive learning algorithm to utilize the structural information and further make use of unlabeled and weakly-labeled data. We use a generative model to leverage the large amount of unlabeled data and introduce path constraints into the learning algorithm to incorporate the structural information of the class hierarchy. The posterior probabilities of both unlabeled and weakly labeled data can be incorporated with path-dependent scores. Since we put a structure-sensitive cost to the learning algorithm to constrain the classification consistent with the class hierarchy and do not need to reconstruct the feature vectors for different structures, we can significantly reduce the computational cost compared to structural output learning. Experimental results on two hierarchical text classification benchmarks show that our approach is not only effective but also efficient to handle the semi-supervised and weakly supervised hierarchical text classification.

CCS CONCEPTS

• **Computing methodologies** → *Semi-supervised learning settings*; Classification and regression trees.

KEYWORDS

Hierarchical Text Classification, Semi-Supervised Learning, Weakly Supervised Learning, Structured Prediction

ACM Reference Format:
Huiru Xiao, Xin Liu, and Yangqiu Song. 2019. Efficient Path Prediction for Semi-Supervised and Weakly Supervised Hierarchical Text Classification. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313658

1 INTRODUCTION

Text classification has always been an important task, particularly with the vast growth of text data on the Web needed to be classified. The applications include news classification [8], product review classification [31], spam detection [25] and so on. Hierarchical classification (HC) and structured prediction are involved since the classes are usually organized as a hierarchy. In recent decades, many approaches have been proposed for HC. For example, top-down classification [34] classifies documents at the top layer and then propagates the results to next layer until the leaves. This greedy strategy propagates the classification error along the hierarchy. Contrarily, bottom-up classification [1] backpropagates the labels from the leaves to the top layer, making the leaves with less training data but sharing some similarities with their parents and siblings may not get well considered and trained. Moreover, structural output learning, such as structural perceptron [6] and structural SVM [37], can leverage the structural information in the class hierarchy well, but they need to do Kesler construction [11, 30] where for each sub-structure, the new features are constructed based on the existing features and the class dependencies. That is why structural output learning usually takes more time to train than top-down and bottom-up approaches. All the above approaches are supervised methods. When there are more unlabeled data, it is more challenging if we consider both class dependencies and efficiency of the practical use of hierarchical text classification.

There exist several ways to use the large amount of unlabeled data, among which semi-supervised learning (SSL) [4] and weakly supervised learning such as dataless classification [3, 33] are two representative ways. An example of SSL is [29]. It uses a mixture multinomial model to estimate the posterior probabilities of unlabeled data, which share the same parameters with the naive Bayes model for the labeled data. More parameters can be introduced to model the hierarchical structure, causing the model redundant and meanwhile not accurate enough. As for weakly supervised learning, dataless classification [33] uses the semantic similarities between label descriptions and document contents to provide weak labels for documents. When applying the weak labels, current approaches simply treat each label similarity independently and do not consider the path constraints in the label hierarchy.

To tackle the above problems for semi-supervised and weakly supervised hierarchical text classification, we propose a path cost-sensitive learning algorithm based on a generative model for text. When estimating the path posterior distribution, the path-dependent scores are incorporated to make the posteriori path-sensitive. The path-dependent score evaluates how accurate the current model is in terms of classifying a document among the paths in the class hierarchy. Then during inference, classification is constrained to keep the consistency of the hierarchy. By this mechanism, we develop a simple model with fewer parameters compared with existing approaches while maintaining the consistency property for the class dependencies in the hierarchy.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313658

The contributions of our paper are as follows:

(1) We propose **a new approach for hierarchical text classification** based on a probabilistic framework. We highlight its meanings in cost-sensitive learning and constraint learning.
(2) We show significant improvements on two widely used hierarchical text classification benchmarks and demonstrate our algorithm's **effectiveness** in **semi-supervised** and **weakly-supervised learning** settings.
(3) Our approach **reduces the complexity** of traditional methods. We achieve tens of times speedup while outperforming the state-of-the-art discriminative baseline.

The code and data used in the paper are available at https://github.com/HKUST-KnowComp/PathPredictionForTextClassification.

2 RELATED WORK

There are only a few studies on semi-supervised hierarchical (text) classification [9, 29], partially because of the difficulty to evaluate the class dependencies for unlabeled data and the time cost of using more complicated algorithms such as structural output learning [23]. Most semi-supervised hierarchical text classification works were based on EM algorithm introduced by [29]. Some are related with ours (see in Section 2.1), while others are not, e.g., [9] used EM algorithm to deal with incomplete hierarchy problem, which was not the same setting as ours. In this section, we simply start with the review of general hierarchical text classification and then explain the uniqueness and significance of our work.

Hierarchical text classification has been studied for several decades. Flat multi-label classification methods [36] ignore the hierarchy, thus poor for HC. Early works [12, 17, 22] often used "pachinko-machine models" which assigned a local classifier at each node and classified documents recursively. Top-down and bottom-up approaches utilize the local classifier ideas, but top-down is a greedy strategy so it may not find optimal solutions, while bottom-up approach does not well consider and train the classes with less training data.

To better exploit the class hierarchy, algorithms particularly designed for trees can assist. In practice, both generative and discriminative models are used. In the following, we will review the related work of these two categories.

2.1 Generative Models

[29] summarized the text generative model and provided the naive Bayes classifier and Expectation-Maximization (EM) algorithm for flat classification. As for HC, it introduced more parameters to account for the class dependencies. [24] remodeled the framework in another way. They applied shrinkage to smooth parameter estimates using the class hierarchy. [7] also used the same generative framework but proposed a clustering-based partitioning technique. These generative hierarchical methods can bring some structural information to the model, but they do not make full use of the hierarchy and have difficulties scaling to large hierarchies.

2.2 Discriminative Models

Discriminative methods are also popular for HC. Orthogonal Transfer [39] borrowed the idea of top-down classification where each node had a regularized classifier and each node's normal vector classifying hyperplane was encouraged to be orthogonal to its ancestors'. Hierarchical Bayesian Logistic Regression [16] leveraged the hierarchical dependencies by giving the children nodes a prior centered on the parameters of its parents. The idea was further developed in Hierarchically Regularized SVM and Logistic Regression [15], where the hierarchical dependencies were incorporated into the parameter regularization structure. More recently, the idea of hierarchical regularization has been applied to deep models and also showed some improvements [32]. [5] simplified the construction of classifier by building a binary classifier on each tree node and providing the cost-sensitive learning (HierCost). All the above approaches are still based on top-down or greedy classification which can result in non-optimal solutions. Another similar work with ours is [38]'s hierarchical loss for classification, which defined the hierarchical loss or win as the weighted sum of the probabilities of the nodes along the path. In contrast to their work, we use the sum of the (weakly) labeled instances along a path as score to perform path cost-sensitive learning.

To find more theoretically guaranteed solutions, some algorithms were developed based on structural output learning [6, 18, 35, 37], which can be proved to be global optimal for HC. Hierarchical SVM (HSVM) [2], one example of structural SVM, generalized SVM to structured data with a path-dependent discriminant function. In general, when performing structural output learning, Kesler construction is used to construct the feature vectors for comparing different structures [11, 30], which adds much more computation than top-down or bottom-up classification approaches.

In summary, generative and discriminative models can both be adapted to HC problems. Discriminative models achieve better performance with adequate labeled data [28], especially if a better representation for text can be found, e.g., using deep learning [32]. Whereas generative models have their advantage for handling more uncertainties [28] for limited labeled data and under noisy supervision. Our work is based on a generative model yet has the same parameter size as the flat classification. We find that it significantly boosts the performance of semi-supervised learning and weakly supervised learning as well as reduces the computational cost.

3 PATH PREDICTION FOR HIERARCHICAL CLASSIFICATION

In HC, the classes constitute a hierarchy, denoted as $\mathcal{T}$. $\mathcal{T}$ is a tree whose depth is d, with the root node in depth 0. Then the classes are distributed from depth 1 to d. We suppose that all leaf nodes are in depth d. This can always be satisfied by expanding the shallower leaf node (i.e. giving it a child) until it reaches depth d. When evaluating models, these dummy nodes from $\mathcal{T}$ can be easily removed to avoid affecting the performance measure.

Let $C_1, \ldots, C_d$ be the class sets in depth $1, \ldots,$ depth d accordingly, with sizes $M_1, \ldots, M_d$. To classify a document, we assign labels in each depth, i.e., the document gets d labels $\{c^k_{j_k} : k = 1, \ldots, d, j_k = 1, \ldots, M_k\}$. These $\{c^k_{j_k}\}$ form a path in $\mathcal{T}$ if the classification results in each depth are consistent with other depths. We want to maintain the consistency of the hierarchy, therefore we classify the documents by paths instead of by multi-label classes. After assigned a path, the document's classes are the nodes lying in the path. It is similar with structured prediction since a path can be regarded as a structured object, which contains more information than a set of multi-label classes without path constraints.

To sum up, path prediction aims at making use of the structural information in the class hierarchy to train the classifier. Note that the classifier is for paths in the hierarchy instead of classes. The details of path prediction algorithm are given in the next section.

4 PATH COST-SENSITIVE LEARNING

In this section, we introduce our method which utilizes the structural information to learn the classifier, revealing its meanings in cost-sensitive learning and constraint learning.

4.1 Path-Generated Probabilistic Framework

We base our work on a widely-used probabilistic framework, which constructs a generative model for text. In the framework, text data are assumed to be generated from a mixture of multinomial distributions over words. Previous works [24, 29] assumed that the mixture components in the generative model have a one-to-one correspondence with the classes. However, in order to perform path prediction, we presume that the mixture components have a one-to-one correspondence with the paths.

Define $\mathcal{P}$ to be the set of all paths which start from the root node and end in some leaf node in the class hierarchy $\mathcal{T}$, so the size of $\mathcal{P}$ equals to that of the leaf nodes M_d. Let $\mathcal{V}$ be the vocabulary. Denote θ as the parameters for the mixture multinomial model. For a document x_i with length $|x_i|$, suppose x_{it} is the word frequency of word w_t in x_i, which is the document feature represented by vector space model [21]. Then the generative process runs as following.

First, select a mixture component, or equivalently a path $p_j \in \mathcal{P}$, from $P(p_j|\theta)$ (prior of p_j). Next, generate the document x_i by selecting the length $|x_i|$ and picking up $|x_i|$ words from $P(x_i|p_j;\theta)$. According to the law of total probability and the naive Bayes assumption that given the labels, the occurrence times of each word in a document are conditionally independent with its position as well as other words, the probability of generating x_i is

$$P(x_i|\theta) \propto P(|x_i|) \sum_{j=1}^{M_d} P(p_j|\theta) \prod_{w_t \in \mathcal{V}} P(w_t|p_j;\theta)^{x_{it}}. \quad (1)$$

In general, document lengths are assumed to be independent with classes, thus independent with paths. So model parameters θ include the path prior $\theta_{p_j} \equiv P(p_j|\theta)$ and the multinomial distribution over words for each path $\theta_{w_t|p_j} \equiv P(w_t|p_j;\theta)$.

4.2 Path-Dependent Scores

Given a data set $\mathcal{D} = \{(x_i, y_i) : x_i \in \mathcal{X}_l\} \cup \{x_i : x_i \in \mathcal{X}_u\}$, consisting of the labeled documents $\mathcal{X}_l$ and the unlabeled documents $\mathcal{X}_u$. We now derive the parameter estimation in a supervised manner. With only labeled data considered, we maximize $P(\theta|\mathcal{X}_l)$, which can be done by counting the corresponding occurrences of events. The event counts are usually the hard counts for flat classification. Here we use a path-dependent score to substitute it.

First we define the score of a node in $\mathcal{T}$ for a document. Suppose $x_i \in X_l$, $c^k_{j_k}$ is the k^{th} node in path p_j. The node score of $c^k_{j_k}$ for x_i, denoted as $S_i(c^k_{j_k})$, indicates the label of x_i. When x_i is labeled with the ground truth labels, $S_i(c^k_{j_k}) = 1$ if and only if $c^k_{j_k}$ is one of x_i's labels. We also consider the weakly supervised case. In [33]'s dataless text classification, for $x_i \in \mathcal{X}_l$, it is weakly labeled by the semantical similarities with classes. We assign value 1 to $S_i(c^k_{j_k})$ if x_i has the largest similarity with $c^k_{j_k}$ among all classes in depth k and 0 otherwise.

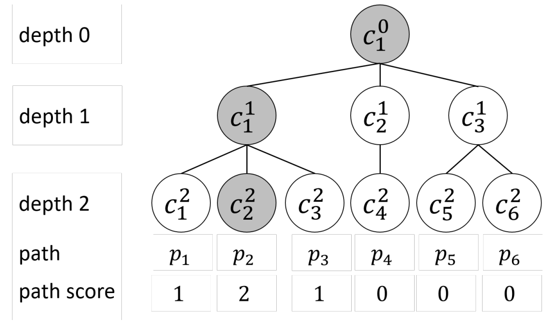


Figure 1: An example of path scores. Suppose x_i is labeled as $\{c^1_1, c^2_2\}$. S_{ij} for $j = 1, \ldots, 6$ is shown in the figure.

Next we introduce the path score. For $x_i \in X_l$, the score of path p_j, denoted as S_{ij}, is the sum of the nodes' scores $S_i(c^k_{j_k})$ lying in p_j except the root node since it makes no sense for classification.

$$S_{ij} \equiv S_i(p_j) = \sum_{k=1}^{d} S_i(c^k_{j_k}). \quad (2)$$

Take the hierarchy in Figure 1 as an example. x_i is labeled as $\{c^1_1, c^2_2\}$, then $S_{i2} = S_i(c^1_1) + S_i(c^2_2) = 2$, $S_{i1} = S_{i3} = 1$, while other paths score 0. If x_i is weakly labeled by the similarities, then we label it with the classes having the maximum similarity in each depth and obtain the path scores in the same way.

4.3 Path Cost-Sensitive Naive Bayes Classifier

While doing the empirical counts, the Laplace smoothing is often applied by adding one count to each event to avoid zero probability and shrink the estimator. Combining the event counts (i.e. the path scores) and the smoothing term, the parameter estimates are:

$$\hat{\theta}_{p_j} \equiv P(p_j|\hat{\theta}) = \frac{1 + \sum_{x_i \in \mathcal{X}_l} S_{ij}}{M_d + \sum_{k=1}^{M_d} \sum_{x_i \in \mathcal{X}_l} S_{ik}}, \quad (3)$$

$$\hat{\theta}_{w_t|p_j} \equiv P(w_t|p_j;\hat{\theta}) = \frac{1 + \sum_{x_i \in \mathcal{X}_l} S_{ij} x_{it}}{|\mathcal{V}| + \sum_{s=1}^{|\mathcal{V}|} \sum_{x_i \in \mathcal{X}_l} S_{ij} x_{is}}. \quad (4)$$

There are two aspects of using the path scores as event counts:

(1) Cost-sensitive performance measures are considered since different data samples are given different weights. In Figure 1, x_i is counted twice for p_2, once in c^1_1 and once in c^2_2, thus obtaining more weights. p_1 and p_3 are not right paths for x_i, but they still classify correctly in depth 1, thus get one count, less than p_2 but larger than other paths who have no correct labels at all. This path **cost-sensitive learning** behavior helps the model to maintain structural information.
(2) The path scores function as the measuring indicators of paths, capacitating the model to classify the documents by paths. The path prediction actually puts constraints on the classifier, where the prediction results must be consistent with the class

hierarchy. Furthermore, the **constraint learning** reduces the search space and improves efficiency.

After estimating θ from $\mathcal{X}_l$, for any test document x_i, the posterior probability distribution can be obtained by Bayes' rule:

$$\begin{aligned} P(y_i = p_j|x_i;\hat{\theta}) &= \frac{P(p_j|\hat{\theta})P(x_i|p_j;\hat{\theta})}{P(x_i|\hat{\theta})} \\ &= \frac{\hat{\theta}_{p_j}\prod_{w_t\in\mathcal{V}}(\hat{\theta}_{w_t|p_j})^{x_{it}}}{\sum_{k=1}^{M_d}\hat{\theta}_{p_k}\prod_{w_t\in\mathcal{V}}(\hat{\theta}_{w_t|p_k})^{x_{it}}}. \end{aligned} \quad (5)$$

Then x_i will be classified into $\arg\max_j P(y_i = p_j|x_i;\hat{\theta})$.

The path cost-sensitive naive Bayes classifier (PCNB) for the generative model are introduced above. Next we will present the semi-supervised path cost-sensitive learning algorithm.

4.4 Semi-Supervised Path Cost-Sensitive Learning

Until now, only the labeled data are used during training, but we want to make use of the unlabeled data to ameliorate the classifier. We follow [29] to apply EM technique for SSL.

When the initial parameters are given, the posterior probabilities of $\mathcal{X}_u$, computed through Eq. (5), can act as the path score S_{ij} for $x_i \in X_u$. Combining the labeled and unlabeled data together, the parameter estimates are changed into

$$\hat{\theta}_{p_j} = \frac{1 + \sum_{x_i\in\mathcal{X}_l\cup\mathcal{X}_u} S_{ij}}{M_d + \sum_{k=1}^{M_d}\sum_{x_i\in\mathcal{X}_l\cup\mathcal{X}_u} S_{ik}}, \quad (6)$$

$$\hat{\theta}_{w_t|p_j} = \frac{1 + \sum_{x_i\in\mathcal{X}_l\cup\mathcal{X}_u} S_{ij}x_{it}}{|\mathcal{V}| + \sum_{s=1}^{|\mathcal{V}|}\sum_{x_i\in\mathcal{X}_l\cup\mathcal{X}_u} S_{ij}x_{is}}. \quad (7)$$

Note that the numerical value of S_{ij} for $x_i \in X_u$ ranges in $[0, 1]$ since it is the posterior probability. Therefore, the unlabeled data weight less than the labeled data while estimating the parameters. It is reasonable because the labeled data are more authentic than the inference results of unlabeled data, especially in the early iterations where the model does not reach convergence.

The new θ obtained via Eqs. (6) and (7) are then used to compute the posterior probabilities of $\mathcal{X}_u$ again, which in turn update θ. The iterative process keeps maximizing the likelihood of the dataset $P(\theta|\mathcal{D})$, equivalent to maximizing the log likelihood:

$$\begin{aligned} l(\theta|X,Y) &\propto log(P(\theta)P(\mathcal{D}|\theta)) = log(P(\theta)) \\ &+ \sum_{x_i\in\mathcal{X}_l} log(P(y_i = p_j|\theta)P(x_i|y_i = p_j;\theta)) \\ &+ \sum_{x_i\in\mathcal{X}_u} log \sum_{p_j\in\mathcal{P}} P(p_j|\theta)P(x_i|p_j;\theta). \end{aligned} \quad (8)$$

Refer to [10], the convergence of EM can be guaranteed, but it reaches some local maxima. To enable the algorithm to find good local maxima, we initialize θ with those obtained through the naive Bayes classifier on $\mathcal{X}_l$. Algorithm 1 presents the EM algorithm for the path cost-sensitive classification (PCEM).

Algorithm 1 Path Cost-Sensitive Algorithm with EM

Input: Training data $\mathcal{D} = \mathcal{X}_l \cup \mathcal{X}_u$, test data $\mathcal{D}_{test} = \mathcal{X}_{test}$, class hierarchy $\mathcal{T}$.

Training:

- Compute path scores for $\mathcal{X}_l$, using Eq. (2).
- Initialize the classifier $\hat{\theta}$ with the path-constrained naive Bayes classifier (Section 4.3).
- **while** $l(\hat{\theta})$ (Eq. (8)) not converged **do**
  E-step: use the current $\hat{\theta}$ to compute the path scores of $\mathcal{X}_u$ with Eq. (5).
  M-step: use the path scores of $\mathcal{X}_l$ and $\mathcal{X}_u$ to re-estimate $\hat{\theta}$, with Eqs. (6) and (7).
- **end while**

Testing:

- Compute the posterior probabilities for $\mathcal{X}_{test}$ with Eq. (5). Classify $\mathcal{X}_{test}$ by selecting $\hat{j} = \arg\max_j P(y_i = p_j|x_i;\hat{\theta})$ for $x_i \in \mathcal{X}_{test}, p_j \in \mathcal{P}$. The classes of x_i are nodes in $p_{\hat{j}}$.

Output: The classifier $\hat{\theta}$ and classification results for $\mathcal{D}_{test}$.

	#Training	#Test	#Features	#Leaves	#Nodes	Depth
20NG	15,077	3,769	103,363	20	27	2
RCV1	6,395	1,733	26,888	35	56	3

Table 1: Statistics of the datasets.

5 EXPERIMENTS

For empirical evaluation of effectiveness and efficiency of our approach, we design experiments on semi-supervised and weakly supervised hierarchical text classification tasks, compared to the representative and the state-of-the-art baselines.

5.1 Experimental Design

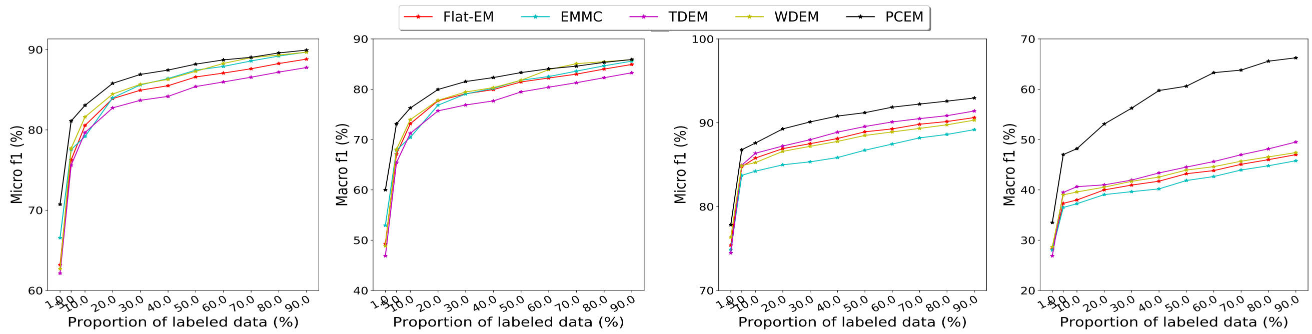
5.1.1 Datasets. We use two datasets, both of which have semi-supervised and weakly-supervised version. The statistics are listed in Table 1.

- **20NG**[1] [19] 20 Newsgroups is a widely-used text classification dataset. To experiment on weakly-supervised setting and compare with semi-supervised baselines, we use dataless 20NG provided in [33].
- **RCV1** [20] RCV1 is a collection of manually labeled Reuters News from 1996-1997. We also use the subset of data provided in [33], which consists of 8,668 single-labeled news documents on the dataless setting.

5.1.2 Baselines. We compare our path cost-sensitive algorithms (**PCNB** and **PCEM**) with the following baselines:

- **Generative baselines**
  (1) Flat naive Bayes classifier (**Flat-NB**) and **Flat-EM** algorithm: the flat classifiers introduced in [29].
  (2) Naive Bayes classifier with multiple components (**NBMC**) and **EMMC**: a more expressive model proposed by [29].
  (3) Top-down naive Bayes classifier (**TDNB**) and **TDEM**: the classifiers run in the top-down way.
  (4) Win-driven naive Bayes classifier (**WDNB**) and **WDEM**: the modified hierarchical loss for classification [38].
- **Discriminative baselines**

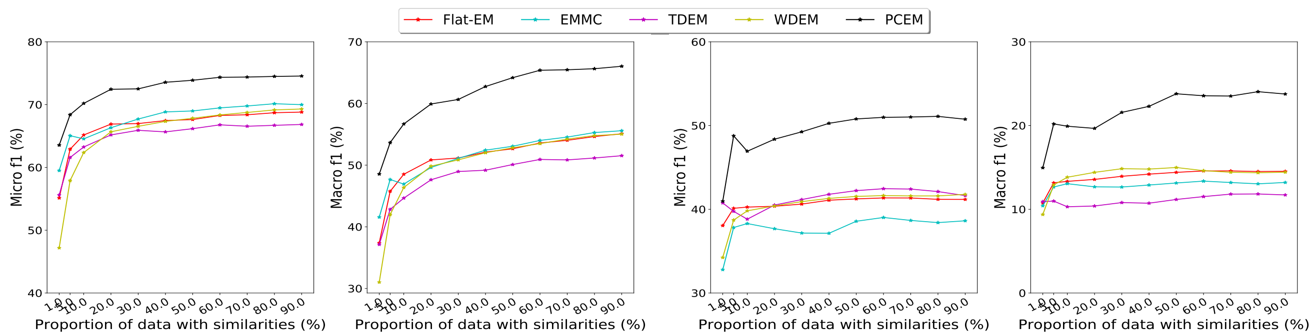
[1] http://qwone.com/~jason/20Newsgroups/



(a) Evaluation Results on 20NG

(b) Evaluation Results on RCV1

Figure 2: Experimental results on semi-supervised classification. (a)-(d) shows the micro-F_1 score and macro-F_1 score on 20NG and RCV1. The horizontal axis is label rate (%). The vertical axis represents the F_1 scores (%).



(a) Evaluation Results on 20NG

(b) Evaluation Results on RCV1

Figure 3: Experimental results on weakly-supervised classification. (a)-(d) shows the micro-F_1 scores and macro-F_1 scores on dataless 20NG and RCV1. The horizontal axis is the proportion of data with semantical similarities (%) in training set. The vertical axis represents the F_1 scores (%).

(1) Logistic regression (**LR**) and **SVM**: two classical discriminative methods. Our experiments use the LibLinear[2] [13] to train corresponding models and test. During the experiments, we found that dual solvers were much faster and even better in performance than primal solvers, so we chose dual solvers.
(2) **HierCost**[3] [5]: the state-of-the-art discriminative method for hierarchical text classification.

5.1.3 Evaluation Metrics. We use F_1 scores [40] to evaluate the performances of all methods. Denote TP_i, FP_i, FN_i as the instance numbers of true-positive, false-positive and false negative for class c_i. Let C be the set of all classes except the root node. Two conventional F_1 scores are defined as:

- **Micro-**$F_1 = \frac{2PR}{P+R}$,
  where $P = \frac{\sum_{c_i \in C} TP_i}{\sum_{c_i \in C}(TP_i+FP_i)}$ is the averaged precision and $R = \frac{\sum_{c_i \in C} TP_i}{\sum_{c_i \in C}(TP_i+FN_i)}$ is the averaged recall.
- **Macro-**$F_1 = \frac{1}{|C|}\sum_{c_i \in C}\frac{2P_iR_i}{P_i+R_i}$,
  where $P_i = \frac{TP_i}{TP_i+FP_i}$ and $R_i = \frac{TP_i}{TP_i+FN_i}$ are the precision and the recall for c_i.

For the two F_1 scores, we measure the overall performance of all classes in the hierarchy in our experiments.

5.2 Results

To evaluate our algorithms, we compare our algorithms with the baselines in semi-supervised and weakly supervised hierarchical text classification. Results on all datasets under 1% label rate are summarized in Table 2, where 1% label rate means there are 1% data in the training set are labeled or weakly-labeled, which is a common setting for semi-supervised text classification. To show that our approach (PCEM) indeed levareges unlabeled data and weakly labeled data well, we present the results under different label rates compared with other EM methods in Figure 2 and 3. For each experiment, we randomly split the training data into labeled and unlabeled according to the label rate, then run experiments using the splitted training data. The running is executed for 5 times and the mean F_1 scores are calculated. Next we will analyze the results. Time efficiency will also be discussed.

[2] https://www.csie.ntu.edu.tw/
[3] https://cs.gmu.edu/~mlbio/HierCost/

	20NG				RCV1			
	labeled		dataless		labeled		dataless	
LR	‡52.02	‡42.41	‡44.16	‡31.51	‡69.59	†24.43	†33.54	‡9.00
SVM	‡48.33	‡39.73	‡41.70	‡30.24	‡68.78	†23.97	†34.15	†9.72
HierCost	‡48.12	‡40.89	‡43.26	‡32.30	‡69.22	†24.98	†31.07	‡8.84
NB	‡53.39	‡39.94	‡47.29	‡30.67	‡70.68	†24.48	†33.29	‡8.39
NBMC	‡46.99	‡38.02	‡43.24	‡28.82	†69.84	†23.52	†28.91	‡6.91
TDNB	‡55.50	‡42.16	‡48.06	‡31.02	‡70.37	†24.65	33.67	†8.40
WDNB	‡53.66	‡41.53	‡47.19	‡31.02	‡70.89	†25.04	34.24	‡9.38
PCNB	‡58.33	‡48.04	‡52.14	†38.50	†73.63	29.95	37.06	12.47
EM	†63.21	†49.30	†55.13	†37.40	75.38	28.32	38.05	†10.76
EMMC	†66.56	†52.95	59.50	†41.56	74.86	28.04	32.79	†10.42
TDEM	†62.14	‡46.89	†55.62	†37.14	†74.48	†26.88	40.76	†10.91
WDEM	†62.71	‡48.85	‡47.19	‡31.02	76.35	28.66	34.24	‡9.38
PCEM	**70.73**	**60.02**	**63.54**	**48.56**	**77.83**	**33.49**	**40.96**	**14.96**

Table 2: F_1 scores (%) on 1% "labeled" data of each dataset. Under each dataset, the two columns are Micro-F_1 (left) and Macro-F_1 (right). The best results are shown in boldface. The statistical significance metrics are marked with either † if p-values < 0.05 or ‡ if p-values < 0.001.

5.2.1 Semi-Supervised Classification with True Labels. Table 2 shows that when the training data are partly labeled with the ground truth labels, PCEM has remarkable superiority over other methods all the time. The discriminative baselines do not have their advantages on the semi-supervised and weakly supervised settings. When compared with generative baselines, our approaches, either naive Bayes (PCNB) or semi-supervised (PCEM), are the best among the corresponding methods. It demonstrates that our algorithms makes good use of the structural information to improve the hierarchical classification.

As expected, EM approaches outperform the corresponding naive Bayes classifier under 1% label rate, which reveals the benefits from the unlabeled data. However, we also noticed that EM may be surpassed by NB when the label rate gets larger. That is related with whether the ratio between labeled and unlabeled data is suitable for SSL, as well as the bias of unlabeled data. This issue has been discussed in previous works [14].

To see the performance in SSL, we compare PCEM with other EM methods in Figure 2. The label rate ranges in $[1.0, 90.0](\%)$. We find that PCEM outperforms others steadily. Other hierarchical EM methods are close to Flat-EM, showing that they takes little advantage of the class hierarchy. The results reveal the effectiveness of PCEM under all label rates for semi-supervised classification.

5.2.2 Weakly-Supervised Classification on Dataless Setting. We also make a comparison with the baselines on dataless text classification. The experimental setting is the same as the semi-supervised classification, except that the training documents do not have labels. Instead, some of them are 'labeled' as classes with the maximal semantical similarities. We use the dataless 20NG and RCV1 datasets provided by [33]. Results are presented in Table 2 and Figure 3.

We find the consistent results with the semi-supervised setting. PCEM can always beat the baselines with significant improvements. PCNB is also better than other NB methods. It is worth noting that the gaps between our algorithms (PCNB and PCEM) and the baselines are bigger than those in the semi-supervised setting. We think the reason is that for this weakly-labeled dataset, the similarities can be seen as noisy labels for documents. In this noisy circumstance, our path cost-sensitive learning algorithm with the probabilistic framework is pretty good at making use of the structural information and features of unlabeled data to recover the true generative distribution.

5.2.3 Efficiency Comparison. Time complexity is also under consideration to evaluate our algorithms. PCNB is highly efficient, faster than all of the other methods except Flat-NB and even competitive with Flat-NB. PCEM is slightly slower than LR and SVM, but that is because EM methods leverage the unlabeled data, which cannot be used by discriminative methods. The trade-off is acceptable, especially considering the excellent performance of PCEM. Furthermore, PCEM also achieves tens of times speedup compared to HierCost.

6 CONCLUSIONS AND FUTURE WORK

We present an effective and efficient approach for hierarchical text classification. Our path cost-sensitive learning algorithm alters the traditional generative model of text with a path-generated model to constrain the classifier by the class hierarchy. We show that our algorithm outperforms other baselines on semi-supervised learning and weakly supervised learning. In addition, our model has the potential of extension to other models, not limited to the generative one, if the path-dependent scores are incorporated appropriately. For the possible future work, we will convert the current framework into a discriminative learning framework following [6] and apply deep neural models to learn a better representation for text [26, 27]. Discrimative framework will further improve the learning when there are more labeled data and deep neural models are more powerful to handle different kinds of weak supervision.

7 ACKNOWLEDGEMENT

This paper was supported by HKUST-WeChat WHAT Lab and the Early Career Scheme (ECS, No. 26206717) from Research Grants Council in Hong Kong.

REFERENCES

[1] Paul N. Bennett and Nam Nguyen. 2009. Refined experts: improving classification in large taxonomies. In *SIGIR*. ACM, 11–18.

[2] Lijuan Cai and Thomas Hofmann. 2004. Hierarchical document categorization with support vector machines. In *CIKM*. ACM, 78–87.

[3] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *AAAI*. AAAI Press, 830–835.

[4] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. 2010. *Semi-Supervised Learning* (1st ed.). The MIT Press.

[5] Anveshi Charuvaka and Huzefa Rangwala. 2015. HierCost: Improving Large Scale Hierarchical Classification with Cost Sensitive Learning. In *ECML/PKDD (1) (Lecture Notes in Computer Science)*, Vol. 9284. Springer, 675–690.

[6] Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP*.

[7] Gao Cong, Wee Sun Lee, Haoran Wu, and Bing Liu. 2004. Semi-supervised Text Classification Using Partitioned EM. In *DASFAA (Lecture Notes in Computer Science)*, Vol. 2973. Springer, 482–493.

[8] Ido Dagan, Yael Karov, and Dan Roth. 1997. Mistake-Driven Learning in Text Categorization. In *EMNLP*. ACL.

[9] Bhavana Dalvi, Aditya Kumar Mishra, and William W. Cohen. 2016. Hierarchical Semi-supervised Classification with Incomplete Class Hierarchies. In *WSDM*. ACM, 193–202.

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.

[11] Richard O Duda and Peter E Hart. 1973. Pattern classification and scene analysis. *A Wiley-Interscience Publication, New York: Wiley, 1973* (1973).

[12] Susan T. Dumais and Hao Chen. 2000. Hierarchical classification of Web content. In *SIGIR*. ACM, 256–263.

[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9 (2008), 1871–1874.

[14] Patrick Fox-Roberts and Edward Rosten. 2014. Unbiased generative semi-supervised learning. *Journal of Machine Learning Research* 15, 1 (2014), 367–443.

[15] Siddharth Gopal and Yiming Yang. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *KDD*. ACM, 257–265.

[16] Siddharth Gopal, Yiming Yang, Bing Bai, and Alexandru Niculescu-Mizil. 2012. Bayesian models for Large-scale Hierarchical Classification. In *NIPS*. 2420–2428.

[17] Daphne Koller and Mehran Sahami. 1997. Hierarchically Classifying Documents Using Very Few Words. In *ICML*. Morgan Kaufmann, 170–178.

[18] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML*. 282–289.

[19] Ken Lang. 1995. NewsWeeder: Learning to Filter Netnews. In *ICML*. Morgan Kaufmann, 331–339.

[20] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5 (2004), 361–397.

[21] Mingyoug Liu and Jiangang Yang. 2012. An improvement of TFIDF weighting in text categorization. *International proceedings of computer science and information technology* (2012), 44–47.

[22] Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. 2005. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations* 7, 1 (2005), 36–43.

[23] Gideon S. Mann and Andrew McCallum. 2008. Generalized Expectation Criteria for Semi-Supervised Learning of Conditional Random Fields. In *ACL*. The Association for Computer Linguistics, 870–878.

[24] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. 1998. Improving Text Classification by Shrinkage in a Hierarchy of Classes. In *ICML*. Morgan Kaufmann, 359–367.

[25] Michael Mccord and M Chuah. 2011. Spam detection on twitter using traditional classifiers. In *international conference on Autonomic and trusted computing*. Springer, 175–186.

[26] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-Supervised Neural Text Classification. In *CIKM*. ACM, 983–992.

[27] Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-Supervised Hierarchical Text Classification. In *AAAI*. AAAI Press.

[28] Andrew Y. Ng and Michael I. Jordan. 2001. On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. In *NIPS*. MIT Press, 841–848.

[29] Kamal Nigam, Andrew McCallum, and Tom Mitchell. 2006. Semi-supervised text classification using EM. *Semi-Supervised Learning* (2006), 33–56.

[30] N. J. Nilsson. 1965. *Learning machines: Foundations of Trainable Pattern-Classifying Systems* (1st ed.). McGraw-Hill.

[31] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *EMNLP*.

[32] Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In *WWW*. ACM, 1063–1072.

[33] Yangqiu Song and Dan Roth. 2014. On Dataless Hierarchical Text Classification. In *AAAI*. AAAI Press, 1579–1585.

[34] Aixin Sun and Ee-Peng Lim. 2001. Hierarchical Text Classification and Evaluation. In *ICDM*. IEEE Computer Society, 521–528.

[35] Benjamin Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-Margin Markov Networks. In *NIPS*. 25–32.

[36] Domonkos Tikk and György Biró. 2004. A hierarchical test categorization approach and its application to FRT expansion. *Austr. J. Intelligent Information Processing Systems* 8, 3 (2004), 123–131.

[37] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research* 6 (2005), 1453–1484.

[38] Cinna Wu, Mark Tygert, and Yann LeCun. 2017. Hierarchical loss for classification. *CoRR* abs/1709.01062 (2017).

[39] Lin Xiao, Dengyong Zhou, and Mingrui Wu. 2011. Hierarchical Classification via Orthogonal Transfer. In *ICML*. Omnipress, 801–808.

[40] Yiming Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Inf. Retr.* 1, 1-2 (1999), 69–90.