On Dataless Hierarchical Text Classification

Yangqiu Song and Dan Roth
Department of Computer Science
University of Illinois at Urbana-Champaign
{yqsong,danr}@illinois.edu

Abstract

In this paper, we systematically study the problem of dataless hierarchical text classification. Unlike standard text classification schemes that rely on supervised training, dataless classification depends on *understanding* the labels of the sought after categories and requires no labeled data. Given a collection of text documents and a set of labels, we show that *understanding* the labels can be used to accurately categorize the documents. This is done by embedding both labels and documents in a semantic space that allows one to compute meaningful semantic similarity between a document and a potential label. We show that this scheme can be used to support accurate multiclass classification without any supervision. We study several semantic representations and show how to improve the classification using bootstrapping. Our results show that bootstrapped dataless classification is competitive with supervised classification with thousands of labeled examples.

Introduction

With the increasing growth of online textual information on the Web, there is an important need to determine the topics of the vast amount of documents that we have around. Many applications, including news classification (Dagan, Karov, and Roth 1997; Joachims 1998), search result organization (Dumais and Chen 2000), online advertising (Agrawal et al. 2013), etc., have placed text categorization as a key problem. In practice, supporting hierarchical categorization is highly preferred since it provides multiple options that vary in their level of abstractions to better fit the context sensitive nature of applications (Chen and Dumais 2000; Dumais and Chen 2000; Sun and Lim 2001; Cai and Hofmann 2004; Liu et al. 2005; Xiao, Zhou, and Wu 2011; Gopal and Yang 2013). This preference is supported by cognitive science studies, indicating that humans favor categorization at multiple levels of abstraction (Murphy 2002).

While the importance of text classification is well recognized, current research has not paid enough attention to the fact that the labels we want to assign to documents are meaningful. Understanding the labels is powerful and can be used to avoid a key bottleneck – the need for labeled data.

Table 1: Comparing supervised and dataless hierarchical text classification on the 20NG dataset. OHLDA refers to an LDA based unsupervised method proposed in (Ha-Thuc and Renders 2011).

	#Labeled data	Best Avg. F_1
Supervised	100	0.515
Supervised	200	0.637
Supervised	500	0.765
Supervised	1,000	**0.825**
Supervised	2,000	**0.866**
OHLDA	0	0.595
Dataless	0	0.682
+ Bootstrapping	0	**0.837**

This was observed first in (Chang et al. 2008), studying a simpler problem of flat binary classification, and later studied in computer vision as "zero-shot learning" (Palatucci et al. 2009; Socher et al. 2013; Elhoseiny et al. 2013). In these settings, label names or descriptions are given instead of labeled data associated to the labels. However, early work has not addressed hierarchical or multi-label classification problems as we do. Moreover, we provide a comprehensive study of what semantic representations best support dataless classification.

Given a collection of text documents and a set of categories, we show that it is possible to assign category labels to documents without requiring any labeled training data. Instead, *understanding* the labels can be used to accurately perform this categorization. Our scheme, *dataless hierarchical text classification* is composed of two steps: a semantic similarity step and a bootstrapping step. In the semantic similarity step, we embed both labels and documents in a semantic space that allows one to compute meaningful semantic similarity between a document and a potential label. While this is a generic step that makes use of external information in the form of the semantic embedding, in the bootstrapping step we adapt to the specific document collection; we use the semantic similarity step to drive a machine learning classifier that iteratively improves the categorization without a need for labeled data. We study dataless classification in the context of two natural hierarchical classification schemes, top-down and bottom-up. Our bottom line results are summarised in Table 1, indicating that dataless classification is competitive with supervised classification with thousands of labeled examples.

Datasets and Evaluation Metrics

Before introducing the algorithms with results, we first introduce the datasets and the evaluation metrics. Ideally, one wants to use a very broad ontology of categories and have the document collection choose which category (or a set of categories) best describes each document in the collection. This is how we envision the use of dataless methods in practice. However, in order to evaluate the quality of our dataless algorithms we need to use existing labeled data. Therefore, we begin by describing the datasets we use to do the evaluation to compare our framework with existing work.

20Newsgroups Data (20NG) The 20 newsgroups data (Lang 1995) is usually used as a multi-class classification benchmark dataset. It contains about 20,000 newsgroups messages evenly distributed across 20 newsgroups. Some of the newsgroups are close to each other so that 20 newsgroups are also categorized into six super-classes, i.e., computers, recreation, religion, science, politics and forsale.[1] We use these two levels of classes as the hierarchical classification problem and, in our evaluation, we aggregate the label descriptions of the 20 newsgroups to the upper level's six super-classes.

Good description of the labels is crucially important for comparing the similarity between labels and documents; nevertheless, we first follow the descriptions of the labels as provided in (Chang et al. 2008). We then provide a new, somewhat embellished, set of descriptions and show the impact this has on the dataless classification performance. The labels and their descriptions are shown in Table 2.

RCV1 Data The RCV1 dataset is an archive of manually labeled newswire stories from Reuter Ltd (Lewis et al. 2004). The news documents are categorized with respect to three controlled vocabularies: industries, topics and regions. We choose to use topics as our hierarchical classification problem. There are in total 804,414 documents. To ease the computational cost of comparison, we choose the 23,149 documents marked as training samples in the dataset. We checked the other four parts of the test data, and the results are similar to the training data. The RCV1 data is a multi-label dataset; that is, a document can belong to several categories. There are 103 categories including all nodes except for root in the hierarchy. The maximum depth is four, and 82 nodes are leaves. The dataset also provides the name and description of each label. For example, label "C11" is named as "strategy plans" with description "strategy, new companies, joint ventures, consortia, diversifications, and investment." We also aggregate all the subtree nodes' descriptions to the root of subtree.

Evaluation Metrics We use averaged F_1 scores to measure the performance of all the methods (Yang 1999). Let TP_i, FP_i, FN_i denote the true-positive, false-positive, and false negative values for the ith label in label set $\mathcal{T}$, when we assign most confident label to each document at each level in the label hierarchy. Then we have the micro-averaged and macro-averaged F_1 scores as: $MicroF_1 = 2\bar{P} * \bar{R}/(\bar{P} + \bar{R})$ and $MacroF_1 = \frac{1}{|\mathcal{T}|}\sum_i^{|\mathcal{T}|}\frac{2P_i * R_i}{P_i + R_i}$, where $P_i = TP_i/(TP_i + FP_i)$ and $R_i = TP_i/(TP_i + FN_i)$ are the precision and recall for ith label, $\bar{P} = \sum_i^{|\mathcal{T}|} TP_i / \sum_i^{|\mathcal{T}|}(TP_i + FP_i)$ and $\bar{R} = \sum_i^{|\mathcal{T}|} TP_i / \sum_i^{|\mathcal{T}|}(TP_i + FN_i)$ are the average precision and recall for all labels in $\mathcal{T}$.

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

Table 2: Description of labels for 20newsgroups data. Old description is used by Chang et al. (2008).

Label	Old Description	New Description
talk.politics.guns	politics guns	gun fbi guns weapon compound
talk.politics.mideast	politics mideast	israel arab jews jewish muslim
talk.politics.misc	politics	gay homosexual sexual
alt.atheism	atheism	atheist christian atheism god islamic
soc.religion.christian	society religion christianity christian	christian god christ church bible jesus
talk.religion.misc	religion	christian morality jesus god religion horus
comp.sys.ibm.pc.hardware	computer systems ibm pc hardware	bus pc motherboard bios board computer dos
comp.sys.mac.hardware	computer systems mac macintosh apple hardware	mac apple powerbook
comp.graphics	computer graphics	graphics image gif animation tiff
comp.windows.x	computer windows x windowsx	window motif xterm sun windows
comp.os.ms.windows.misc	computer os operating system microsoft windows	windows dos microsoft ms driver drivers card printer
rec.autos	cars	car ford auto toyota honda nissan bmw
rec.motorcycles	motorcycles	bike motorcycle yamaha
rec.sport.baseball	baseball	baseball ball hitter
rec.sport.hocke	hockey	hockey wings espn
sci.electronics	science electronics	circuit electronics radio signal battery
sci.crypt	science cryptography	encryption key crypto algorithm security
sci.med	science medicine	doctor medical disease medicine patient
sci.space	science space	space orbit moon earth sky solar
misc.forsale	for sale discount	sale offer shipping forsale sell price brand obo

Dataless Hierarchical Classification

In the simplest sense, a dataless classification performs a nearest neighbor classifier in an appropriately selected semantic feature space (Chang et al. 2008). Let $\phi(l_i)$ be the vector for the semantic representation of label l_i, and $\phi(d)$ the representation of document d. We select that category $l_i^* = \arg\min_i ||\phi(l_i) - \phi(d)||$. When a label hierarchy is available, we can reduce the classification complexity by considering the structure of hierarchy. Then, with available unlabeled data, we can further improve classification using bootstrapping. In this section, we first introduce several possible semantic representation approaches for text, and then show how to design a dataless hierarchical classifier.

The Importance of Semantic Representation

The semantic representation of textual content is one of the most important issues for text classification. Traditional supervised classification makes use of the bag-of-words (BOW) representation of documents. It breaks a document into words, and formalizes the term frequency (TF) or the term frequency-inverse document frequency (TFIDF) scores of words as a high-dimensional sparse vector. Then, taking linear kernel support vector machine (SVM) as an example, the classifier uses a linear combination of training data as the weight vector, to represent a classification hyperplane.

In the dataless setting, we do not assume the availability labeled data. We only assume that we have the names of labels. Directly comparing the BOW representations of the documents with the label description may not be sufficient

due to sparsity. For example, a news article discussing *sport* may only mention names of players, teams, or activities of a match without mentioning the word *sport*. Therefore, finding a better semantic representation is essential for dataless classification. In this section, we propose several possible representations.

Explicit Semantic Analysis (ESA) Explicit semantic analysis (ESA) uses Wikipedia as an external knowledge base to generate *concepts* for a given fragment of text (Gabrilovich and Markovitch 2006; 2007). It assumes that each Wikipedia article corresponds to a concept, and uses the title of the article as the name of the concept.

ESA first represents a given text fragment as a TFIDF vector, then uses an inverted index for each word to search the Wikipedia corpus, and finally merges the retrieved concepts weighted by the TFIDF scores of words (Gabrilovich and Markovitch 2006; 2007). The text fragment representation is thus a weighted combination of the concept vectors corresponding to its words. We implemented ESA using the latest dump of Wikipedia, which contains about 13 millions pages, including redirection and disambiguation pages. After filtering out pages with less than 100 words and those containing less than 5 hyperlinks, we finally obtain 3.1 million concepts. To evaluate the effectiveness of this concept representation, we use for each text fragment representations of size 50, 100, 200, 500, and 1,000 concepts.

While ESA has been used before for text categorization, we propose to study its effectiveness in dataless classification in comparison to two additional semantic representations, using Brown clusters of words and using neural network embedding of words, discussed next.

Brown Clusters The Brown clusters of words was proposed by Brown et al. (1992) as a way to support abstraction in NLP tasks; it was further used in several NLP works, such as by Liang (2005), to measure words' distributional similarity. This method generates a hierarchical tree of word clusters by evaluating the word co-occurrence based on an n-gram model. Then, paths traced from root to leaves can be used as word representations. We use the implementation by Liang (2005) and generated Brown clusters of words using three corpora in different settings:

$\mathrm{BC}_{\mathrm{20NG}}$: We first performed the Brown clustering on the 20NG data, and set the maximum number of clusters to 50, 100, 200, 500, and 1,000.

$\mathrm{BC}_{\mathrm{RCV1}}$: We also use the Brown clusters from Ratinov and Roth (2009) and Turian, Ratinov, and Bengio (2010).[2] The Brown clusters are generated based on the RCV1 corpus, and the maximum numbers of clusters are set to 100, 320, 1,000, and 3,200.

$\mathrm{BC}_{\mathrm{Wiki}}$: Finally, for a fair comparison with ESA, we ran Brown clustering over the latest Wikipedia dump, and set the maximum number of clusters to 50, 100, 200, and 500.

Neural Network Word Embedding Word embedding trained by neural networks has been used widely in the NLP community and has become a hot trend recently. In this paper, we test the suitability of several different word embeddings for dataless classification.

$\mathrm{WE}_{\mathrm{Senna}}$: We downloaded the word embedding used by Senna neural network (Collobert et al. 2011).[3] The word embedding is trained using an earlier dump of Wikipedia, and results in a 50-dimension embedding of words.

$\mathrm{WE}_{\mathrm{Turian}}$: This is the embedding trained by Turian, Ratinov, and Bengio (2010) over the RCV1 data. The dimensions of word vectors are 25, 50, 100, and 200.

$\mathrm{WE}_{\mathrm{Mikolov}}$: We finally used the tool released by Mikolov, Yih, and Zweig (2013) and Mikolov et al. (2013) over the latest Wikipedia dump, resulting in word vectors with dimensions 50, 100, 200, 500, and 1,000.

Given the above semantic representation of words, we follow the scheme used for ESA to generate the document semantic representation using a TFIDF weighted combination of word vectors in the documents. Note that the Brown clusters obtained from 20NG data ($\mathrm{BC}_{\mathrm{20NG}}$) and RCV1 data ($\mathrm{BC}_{\mathrm{RCV1}}$), and the word embedding of RCV1 data ($\mathrm{WE}_{\mathrm{Turian}}$) should not be available as external knowledge if we want to have a pure dataless setting.

Dataless Hierarchical Classification Algorithms

Our dataless scheme for hierarchical classification consists of two steps: the first is an initial, "pure", dataless classification, and the second performs bootstrapping.

Pure Dataless Initialization Given the external knowledge of word representation, the most basic idea is to perform dataless hierarchical classification by comparing the semantic representations of a document and candidate labels. This primitive step can be run in a top-down or bottom-up way, if one wants to consider the structure of the hierarchical label tree. A top-down algorithm starts from the root node, and greedily finds the best children to further compare. A bottom-up algorithm first compares all the leaves nodes in the tree, and propagates the labels with high confidence scores to the root. Summaries of the algorithms are shown in Algorithms 1 and 2. The threshold δ shown in the algorithms is empirically set to be 0.95. Note that while in standard classification schemes it is essential to go top-down due to lack of supervised data (Chen and Dumais 2000; Dumais and Chen 2000; Sun and Lim 2001; Cai and Hofmann 2004; Liu et al. 2005; Xiao, Zhou, and Wu 2011; Gopal and Yang 2013), in dataless, which builds on "understanding" the labels and the documents, it is not essential, especially in cases when the meaning of a leaf node is quite well distinguished from other labels.

Dataless + Bootstrapping Inspired by the dataless flat classification paper (Chang et al. 2008), we also propose a bootstrapping procedure for dataless hierarchical classification. This is a natural step to follow since it is free (no labeled data is needed) and it provides the dataless algorithm a way to weigh the generic semantic representation in a way that best fits the specific data collection. The bootstrapping step makes use of unlabeled data (the given document collection or additional unlabeled data if so desired), and it labels the most confident documents in each iteration, starting with the labels given in the "pure" initial step. Then, it train-

[2] http://metaoptimize.com/projects/wordreprs/

[3] http://ml.nec-labs.com/senna/

Algorithm 1 Top-down Pure Dataless HC.

Input Data: A hierarchy of label tree $\mathcal{T}$. A representation mapping function $\phi_x(\cdot)$, where x can be ESA, BC, or WE. A document d for classification.
Input Parameters: Cutoff threshold δ. Top K labels at each level.
Initialization: For each node $l \in \mathcal{T}$, get $\phi_x(l)$. Let $l = root$. Output label set $\mathcal{L} = \varnothing$.
Call $TD(d, l)$ as:
if $|\text{children}(l)| > 0$ **then**
  for all $l_i \in \text{children}(l)$ **do**
    $s_i = \cos(\phi_x(l_i), \phi_x(d))$
  end for
  Sort $s_1, ...s_{|\text{children}(l)|}$ as $s'_1, ...s'_{|\text{children}(l)|}$ in descent order and normalize as $\sum_i s'_i = 1$.
  for $i = 1, ..., |\text{children}(l)|$ **do**
    $s_0 \leftarrow s_0 + s'_i$
    if $s_0 > \delta$ **then**
      break
    end if
    $\mathcal{L} \leftarrow \mathcal{L} \cup (l_i, s'_i)$
    Call $TD(d, l_i)$
  end for
end if
Output: Sort the labels in $\mathcal{L}$ at each level (depth), and reports the top K labels at each level.

Algorithm 2 Bottom-up Pure Dataless HC.

Input Data: A hierarchy of label tree $\mathcal{T}$. A representation mapping function $\phi_x(\cdot)$, where x can be ESA, BC, or WE. A document d for classification.
Input Parameters: Cutoff threshold δ. Top K labels at each level.
Initialization: For each node $l \in \text{leaves}(\mathcal{T})$, get $\phi_x(l)$. Output label set $\mathcal{L} = \varnothing$.
for all $l_i \in \text{leaves}(\mathcal{T})$ **do**
  $s_i = \cos(\phi_x(l_i), \phi_x(d))$
end for
Sort $s_1, ...s_{|\text{leaves}(\mathcal{T})|}$ as $s'_1, ...s'_{|\text{leaves}(\mathcal{T})|}$ in descent order and normalize as $\sum_i s_i = 1$.
for $i = 1, ..., |\text{leaves}(\mathcal{T})|$ **do**
  $s_0 \leftarrow s_0 + s'_i$
  if $s_0 > \delta$ **then**
    break
  end if
  $\mathcal{L} \leftarrow \mathcal{L} \cup (l_i, s'_i)$
  for all Ancestors l_a of l_i **do**
    $\mathcal{L} \leftarrow \mathcal{L} \cup (l_a, s'_i)$
  end for
end for
Output: Sort the labels in $\mathcal{L}$ at each level (depth), and reports the top K labels at each level.

s a new classifier to improve its accuracy and incorporate more labeled data. The procedure is as follows:

Step 1: Initialize N documents for each label, using confident pure dataless classifications.

Step 2: For each iteration, train a hierarchical classifier based on BOW representation to label N more documents for each label.[4]

Step 3: Continue until no unlabeled documents remain.

To compare with the dataless scheme, we also implemented supervised hierarchical classifiers in top-down and bottom-up fashions. For the top-down approach, we train a multi-class classifier for each node in the hierarchy tree. As in the dataless case, it greedily searches for the best top K children for further classification. For the bottom-up approach, we train a multi-class classifier for all the leaves nodes, and then propagate the labels to the root. The supervised classification proceeds in exactly the same manner done during bootstrapping, with the only difference that in the supervised case the labels are the gold labels given, and in the dataless scheme the labels are provided by previous bootstrapping stages.

Experiments

Our experiments are designed to study the effectiveness of dataless hierarchical classification in comparison to "standard" supervised classification algorithms, and to study the contribution of different semantic representations to the success of the dataless scheme.

Evaluation of Semantic Representation We first compare the different semantic representations mentioned before, using pure dataless top-down and bottom-up algorithms. The results of $MicroF_1$ scores are shown in Fig. 1.

[4]Our experiments show that bootstrapping with BOW features is the best choice among the different semantic representations.

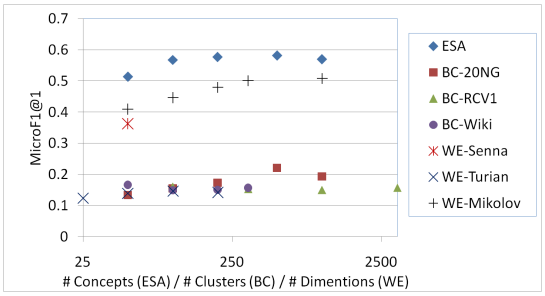
It is evident that the performance of the ESA representation is, by far, the best for both datasets and both top-down and bottom-up algorithms. In general, with more concepts in the representation, the ESA classification results are better.

For word embedding approaches, $\text{WE}_{\text{Mikolov}}$, which is trained on Wikipedia, results in better $MicroF_1$ scores than WE_{Senna} and $\text{WE}_{\text{Turian}}$ but, still, significantly inferior to ESA. WE_{Senna} is also trained on Wikipedia, but the embedding only has 50 dimensions, and it is trained on an earlier Wikipedia dump, which might explain the better performance of $\text{WE}_{\text{Mikolov}}$ for the same dimensionality. For $\text{WE}_{\text{Turian}}$, which is trained on RCV1 data, it is clear that the results on 20NG data are much worse than WE_{Senna} and $\text{WE}_{\text{Mikolov}}$, but comparable to $\text{WE}_{\text{Mikolov}}$ on RCV1.

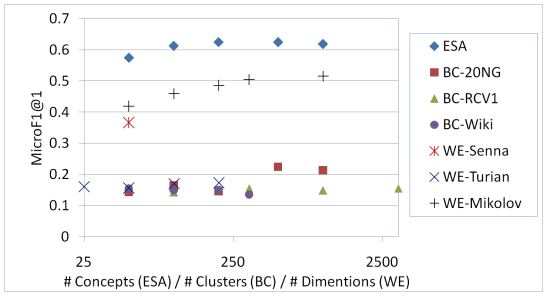
All the Brown cluster based representations do not show promising results on the two datasets. This may be because Brown cluster uses a hierarchial representation of word clusters. When we aggregate all the words' clusters into a document representation, it may become less discriminative since multiple documents might share a lot of common word clusters near the root. More experiments might be needed to better understand this. However, we can still see that BC_{20NG} performs better than the other two Brown cluster based representations on 20NG data, while BC_{RCV1} performs better on RCV1 data.

Unsupervised Baseline In order to demonstrate the quality of our dataless approach, we compare it to some existing unsupervised hierarchical text classification; specifically the recent work on Ontology Guided Hierarchical Latent Dirichlet Allocation (OHLDA). We implemented OHLDA (Ha-Thuc and Renders 2011) for the sake of this experiment. The original OHLDA retrieves 50 documents using a general search engine and 10 documents from Wikipedia for each label, and trains a hierarhical topic model using this data. Here we only retrieve from Wikipedia, but make use of a much larger set of documents, 100 and 500, for each label.

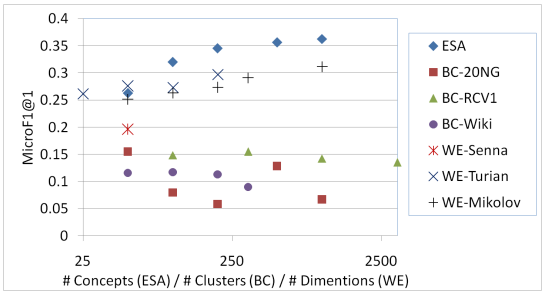
The results of OHLDA are shown in Table 3. Because



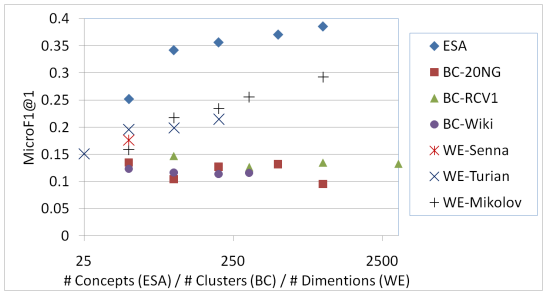
(a) 20newsgroups: top-down (BOW: 0.200)



(b) 20newsgroups: bottom-up (BOW: 0.251)



(c) RCV1: top-down (BOW: 0.279)



(d) RCV1: bottom-up (BOW: 0.276)

Figure 1: $MicroF_1$@1 results of different representations for pure dataless hierarchical classification. "ESA" represents the method using explicit semantic analysis. "BC" represents the methods using Brown clusters. "WE" represents the methods using word embedding. "BOW" represents the classification using the bag-of-words representation. It is clear the ESA is the best semantic representation in all conditions.

OHLDA does not support a bootstrapping step, we compare it here only with the initial, pure, dataless process. Even this way, our dataless approach is clearly superior. It is interesting to observe that using more documents may not improve OHLDA's accuracy: for both dataset, using 100 retrieved Wikipedia pages outperforms the results using 500 pages. We believe that using a larger number of retrieved documents introduces that the topic models cannot tolerate. Moreover, since 20NG has 26 labels and RCV1 has 103 labels, training OHLDA with 100 Wikipedia articles per label requires a large number of documents. Indeed, training and testing (prediction with new data) using OHLDA is significantly more time consuming than all the other algorithms.

Table 3: Comparison OHLDA and the initial (pure) dataless version, as a function of different numbers of retrieved Wikipedia pages for OHLDA. The numbers of document are per label and the results are averaged over ten trails. For 20NG data, we provide the ESA results with both old and new label descriptions ("old/new").

20newsgroups	Pure Dataless	100	500
$MicroF_1$	**0.625/0.682**	0.595±0.001	0.574±0.001
$MacroF_1$	**0.502/0.596**	0.479±0.002	0.463±0.001
RCV1	Pure Dataless	100	500
$MicroF_1$	**0.371**	0.284±0.004	0.274±0.003
$MacroF_1$	**0.183**	0.114±0.002	0.115±0.002

Supervised Baselines To further demonstrate the effectiveness of dataless hierarchical classification, we compared it with several supervised baselines. We implement two framework of supervised models, top-down and bottom-up hierarchical classifiers. The bottom-up mechanism is sometimes called flat classification in the literature (Xiao, Zhou, and Wu 2011; Gopal and Yang 2013).

Naive Bayes (NB): We first use a NB classifier for each node, since the original dataless classification work has compared with naive Bayes (Chang et al. 2008). The NB classifier is implemented using LBJava (Rizzolo and Roth 2010). The features used by this implementation are binary values of words. For each word in the vocabulary, if a document contains the word, then the value on that dimension is set to be one, otherwise, zero.

Logistic Regression (LR): We also incorporate LR in the supervised framework which has been used by Gopal and Yang (2013). The implementation of logistic regression is based on Liblinear (Fan et al. 2008). We choose the one-vs-rest multi-class classification with L2-regularization approach. The features used are TFIDF vector of words. The IDF score is computed based on the training data.

Support Vector Machine (SVM): Hierarchical SVM has been used in many papers (Dumais and Chen 2000; Cai and Hofmann 2004; Liu et al. 2005; Gopal and Yang 2013). We use the Crammer & Singer's pairwise class comparison approach which is implemented in Liblinear (Fan et al. 2008). The features used in SVM are the same as in LR.

The results of the supervised methods for 20NG and RCV1 data are shown in Figs. 2 and 3.

We can see that among the supervised methods, SVM performs the best for 20NG. For RCV1, the number of classes is larger and the number of examples each class is smaller. In this case, one-vs-rest approaches, i.e., NB and LR, perform better than SVM's pairwise approach. Moreover, NB, with binary features, seems to be more stable in this case of large number of classed with few examples each. It is also interesting to see that, for 20NG, bottom-up is better than top-down mechanism, but when the number of labels increases, as in RCV1, the top-down mechanism seems to be more sta-

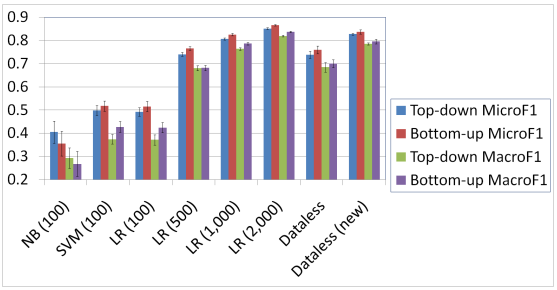


Figure 2: 20newsgroups: comparison of dataless hierarchical classification with supervised baselines. All methods are evaluated based on average of ten randomly sampled trials. "SVM (100)" represents SVM with 100 labeled data. "Dataless" means ESA (500) + bootstrapping. "Dataless (new)" means ESA (500) with new descriptions (in Table 2) + bootstrapping.

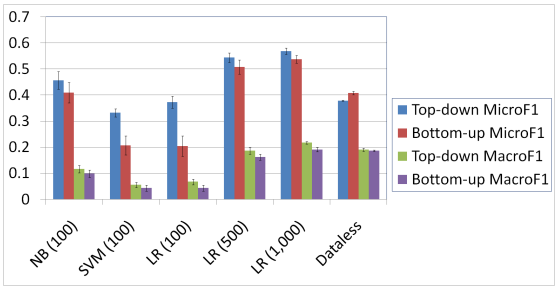


Figure 3: RCV: comparison of dataless hierarchical classification with supervised baselines. All methods are evaluated based on average of ten randomly sampled trials. "SVM (100)" represents SVM with 100 labeled data. "Dataless" means ESA (500) + bootstrapping.

ble and shows higher F_1 scores. The reason is that in higher levels of the hierarchy top-down has less class labels to work with, and thus has more examples for each class.

Dataless Classification Our key experiments make use of the two step datalass classification process: we choose ESA with 500 concepts as the initialization approach, and use L-R as the base classifier for each node, since we find it to be more stable across datasets. In the bootstrapping process, when we find a labeled documents (via previous steps), we also add the label to all the ancestors' labeled sets. Therefore, for each iteration, there will be more than $N * |\mathcal{T}|$ documents added in the tree of classifiers. For 20NG data, we randomly sample 50% of the document set and allow the bootstrapping process to access it, and we use the rest as test data. For RCV1, bootstrapping can access 80% of the documents for training, for compatibility with the supervised methods. We empirically set $N = 20$ for both datasets. All the results are average of ten trials.

The bottom line is that the results of the overall dataless process (Figs. 2 and 3) is shown to be competitive with supervised training. Specifically, for 20NG, bootstrapping is competitive with supervised methods with 500 labeled documents for old description, and with 1,000 labeled documents for new label descriptions. For RCV1 data, although the performance of dataless classification is worse than supervised algorithms on $MicroF_1$ scores with top-down approaches, it is shown that dataless approaches are significantly better on $MacroF_1$. The reason is that for RCV1, there are several classes with no training data. While this does not impact the dataless approach, it does affect the average result over all labels of the supervised approaches. The results of dataless on $MacroF_1$ are also competitive with supervised methods with 500 labeled documents.

Discussion and Conclusion

In this section we discuss some further analysis we conducted to better understand our results and assess the practical use of dataless classification. The scenario we envision for dataless classification includes a collection of documents along with an ontology of possible categories that we want to assign to each document. While for evaluation purpose, we had to work with a small and closed set of category labels, we believe that this type of evaluation does not reflect the true ability of dataless classification. To validate this intuition, we performed two additional experiments.

First, we renamed the categories in the 20NG dataset to better reflect the content of the collection (shown in Table 2). Given the new descriptions, we tested some of the semantic representations and compared them with the previous performance. The key observation is that the dataless classification given by both ESA and $\mathrm{WE}_{\mathrm{Mikolov}}$ are significantly improved. As a consequence, the bootstrapping results are also improved (results of ESA are shown in Table 3 and Fig. 2).

In our second experiment we used the Yahoo! Directory's categories as the dataless labels. We used 661 unique categories which are the leaves in our hierarchy, taken from the first, second and (some of the) third level of the hierarchy. Once we classified the 20NG documents into this large hierarchy, we analyzed the result by comparing the labels given by the dataless algorithm to the gold labels. The results are very satisfying. For example, the documents in the "rec.autos" newsgroup are mostly classified to Yahoo! categories "news and media: traffic and road conditions" and "sports: wheelchair racing." Moreover, documents in newsgroup "talk. politics.misc" that are known to contains document on social issues are classified mostly into Yahoo! categories "news and media: cultures and groups," "social science: lesbian gay bisexual and transgendered studies," "health: long term care," etc. Finally, we observe that coherent groups are classified as such – most of documents classified in "science: aeronautics and aerospace" are from "sci.space" newsgroup. Our conclusion is that given a large label hierarchy such as the Yahoo! Directory, our dataless method allows for robust organization of the documents by their content.

Overall, we proposed a dataless hierarchical classification approach for text categorization. Hierarchical classification is a more general and realistic protocol for text classification. We studied both top-down and bottom-up mechanisms and showed that "bottom-up" approache is more useful in the dataless setting. We systematically compared the ESA approach to other "modern" representations, i.e., Brown clusters, word embedding, and OHLDA topics, thus demonstrating the importance of representation for dataless classification. Not surprisingly, ESA is found to be better suited for this task. Finally, our experiments indicate that dataless hierarchical classification is a promising and practical direction.

Acknowledgements

This work is supported by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20155, and by DARPA under agreement number FA8750-13-2-0008. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied by these agencies or the U.S. Government.

References

Agrawal, R.; Gupta, A.; Prabhu, Y.; and Varma, M. 2013. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, 13–24.

Brown, P. F.; Pietra, V. J. D.; DeSouza, P. V.; Lai, J. C.; and Mercer, R. L. 1992. Class-based n-gram models of natural language. *Computational Linguistics* 18(4):467–479.

Cai, L., and Hofmann, T. 2004. Hierarchical document categorization with support vector machines. In *CIKM*, 78–87.

Chang, M.; Ratinov, L.; Roth, D.; and Srikumar, V. 2008. Importance of semantic representation: Dataless classification. In *AAAI*, 830–835.

Chen, H., and Dumais, S. 2000. Bringing order to the web: Automatically categorizing search results. In *CHI*, 145–152.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12:2493–2537.

Dagan, I.; Karov, Y.; and Roth, D. 1997. Mistake-driven learning in text categorization. In *EMNLP*, 55–63.

Dumais, S., and Chen, H. 2000. Hierarchical classification of web content. In *SIGIR*, 256–263.

Elhoseiny, M.; Saleh, B.; ; and A.Elgammal. 2013. Write a classifier: Zero shot learning using purely textual descriptions. In *ICCV*, 1433–1441.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.* 9:1871–1874.

Gabrilovich, E., and Markovitch, S. 2006. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, 1301–1306.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, 1606–1611.

Gopal, S., and Yang, Y. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. In *KDD*, 257–265.

Ha-Thuc, V., and Renders, J.-M. 2011. Large-scale hierarchical text classification without labelled data. In *WSDM*, 685–694.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In *ECML*, 137–142.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *ICML*, 331–339.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.* 5:361–397.

Liang, P. 2005. Semi-supervised learning for natural language. Master's thesis, Massachusetts Institute of Technology.

Liu, T.-Y.; Yang, Y.; Wan, H.; Zeng, H.-J.; Chen, Z.; and Ma, W.-Y. 2005. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.* 7(1):36–43.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 746–751.

Murphy, G. L. 2002. *The Big Book of Concepts*. MIT Press.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *NIPS*, 1410–1418.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 147–155.

Rizzolo, N., and Roth, D. 2010. Learning based java for rapid development of NLP systems. In *LREC*.

Socher, R.; Ganjoo, M.; Manning, C. D.; and Ng, A. Y. 2013. Zero-shot learning through cross-modal transfer. In *NIPS*, 935–943.

Sun, A., and Lim, E.-P. 2001. Hierarchical text classification and evaluation. In *ICDM*, 521–528.

Turian, J.; Ratinov, L.; and Bengio, Y. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*, 384–394.

Xiao, L.; Zhou, D.; and Wu, M. 2011. Hierarchical classification via orthogonal transfer. In *ICML*, 801–808.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Inf. Retr.* 1(1-2):69–90.