Concept Labeling: Building Text Classifiers with Minimal Supervision

Vijil Chenthamarakshan, Prem Melville, Vikas Sindhwani and Richard D. Lawrence
{ecvijil, pmelvil, vsindhw, ricklawr}@us.ibm.com
IBM T.J Watson Research Center, Yorktown Heights, NY 10598

Abstract

The rapid construction of supervised text classification models is becoming a pervasive need across many modern applications. To reduce human-labeling bottlenecks, many new statistical paradigms (e.g., active, semi-supervised, transfer and multi-task learning) have been vigorously pursued in recent literature with varying degrees of empirical success. Concurrently, the emergence of Web 2.0 platforms in the last decade has enabled a world-wide, collaborative human effort to construct a massive ontology of concepts with very rich, detailed and accurate descriptions. In this paper we propose a new framework to extract supervisory information from such ontologies and complement it with a shift in human effort from direct labeling of examples in the domain of interest to the much more efficient identification of concept-class associations. Through empirical studies on text categorization problems using the Wikipedia ontology, we show that this shift allows very high-quality models to be immediately induced at virtually no cost.

1 Introduction

The explosion of user-generated content by way of blogs and Twitter has given rise to a host of different applications of text categorization, collectively referred to as Social Media Analytics [Melville *et al.*, 2009], to glean insights from this sea of text. The very dynamic nature of social media presents the added challenge of requiring many classifiers to be built on the fly, e.g., building a classifier to identify relevant tweets on the latest smartphone fad, which may be critical for Marketing and PR. As performance of automatic text categorization methods is gated by the amount of supervised data available, there have been many directions explored to get the most out of the available data and human effort. These include (1) exploiting unlabeled data through semi-supervised learning [Chapelle *et al.*, 2005], (2) having the learner select informative examples to be labeled via active learning [Settles, 2009], (3) alternative forms of supervision, such as labeling features [Druck *et al.*, 2008], (4) learning from data in related domains through transfer learning [Blitzer *et al.*, 2007], and (5) guided learning, where human oracles use their domain expertise to seek instances representing the interesting regions of the problem space [Attenberg and Provost, 2010]. All of these approaches still rely on human experts providing labels for *individual* examples or features, and improve with more labels. In this paper we propose an approach to highly scalable supervision, where a very small fixed amount of human effort can be translated to supervisory information on many unlabeled examples, at no additional cost.

Our approach to scalable supervision is enabled by the staggering growth in knowledge-bases and ontologies, generated through collective human effort or semi-automatic processes, such as Wikipedia, Word Net and the Gene Ontology. While these ontologies were not constructed with a specific classification task in mind, the vast amounts of domain-specific and/or general knowledge can still be exploited to improve the way we build supervised models for a given task. In the traditional supervised learning paradigm, supervisory information is provided by labeling examples, and classifiers are induced using such labeled examples. In this paper we propose a shift to *Concept Labeling,* where instead of labeling individual examples, we provide a mapping between concepts in an ontology to the target classes of interest. The process of mapping unlabeled documents (examples) into concepts in an ontology can be fully-automated, e.g., mapping keywords in a document to corresponding Wikipedia entries [Ferragina and Scaiella, 2010]. Hence, such a mapping requires no additional human labor. Thus instead of labeling individual documents, human effort is better spent on simply labeling concepts in the ontology with the classes of interest, e.g. mapping the Wikipedia categories *oncology* and *anatomical pathology* to the medical publication class on *neoplasm.*

Since most unlabeled documents can be automatically mapped to concepts in a given ontology, we can use the few provided concept labels to then automatically label available unlabeled documents. All of this comes at a fixed, one-time cost of providing ontology-to-class mappings via concept labels. Once we automatically generate ontology-based labeled documents, we are free to apply any text categorization method of choice to build a classifier that generalizes to unseen (test) documents. Concept Labeling should not be confused with previous approaches to using ontologies in classification, which have focused on enhancing the existing instance representation with new ontology-based features [Gabrilovich and Markovitch, 2006]. Instead, we are proposing an alternative use of human annotation effort in labeling concepts in an ontology, which we demonstrate is more cost-effective than labeling documents, and induces higher accuracy classifiers than several other approaches.

2 Concept Labeling Framework

Let us begin by recalling the familiar text categorization setting. A large number of documents, $\{d_i\}_{i=1}^n$, is typically collected by an automated process such as a web crawler. Given a document d, we assume that there is an unknown true conditional distribution $P(y|d)$ over binary categories, $y \in \{-1, 1\}$[1]. By human annotation effort, a small subset of documents are labeled by sampling $y_i \sim P(y|d_i), i = 1 \ldots l$, where the number of labeled documents, l, is typically much smaller than the total number of documents collected. Next, a representation for documents is chosen. Let $\psi_{bow}(d)$ represent the popular bag-of-words representation for document d. A supervised learning model can now be set up as a proxy for the underlying true distribution. Such a model may broadly be specified as follows,

$$P(y|d) = P(y|\psi_{bow}(d), \alpha) \tag{1}$$

where the model parameters α are tuned to fit the labeled examples while being regularized to avoid overfitting. The dominant cost and the primary bottleneck in this end-to-end process is the collection of human labeled data.

We contrast this traditional process with a new framework that assumes the availability of an ontology $\mathcal{O} = (V, E, \psi_{ont})$ which we formalize in terms of a triplet: (i) a set of concepts V, (ii) a graph of directed edges E that captures interrelationships between concepts, i.e., an edge $(v_1, v_2) \in E$ indicates that v_2 is a sub-concept of v_1, and (iii) a feature function ψ_{ont} that associates each concept in V to a set of numerical attributes. We will shortly make this specification concrete for the Wikipedia ontology in our text categorization system. We now assume that *categories are conditionally independent of documents, given the concepts of the ontology*. In other words, in comparison to Eq. 1, we instead have,

$$P_{ont}(y|d) = \sum_{v \in V} P(y, v|d) = \sum_{v \in V} P(v|d)P(y|v, \beta) \tag{2}$$

We refer to $P(v|d)$ as the *Documents-to-Ontology* distribution, and to $P(y|v, \beta)$ as the *Ontology-to-Class* distribution. These distributions are modeled separately in our framework and take the graph structure of the ontology into account. We propose an unsupervised construction of the documents-to-ontology distribution, but a supervised construction of the ontology-to-class distribution. In other words, we require human effort to instead be expended in supplying a labeled set $\{v_i, y_i\}_{i=1}^l$ where $y_i \sim P(y|v_i)$. The model parameters β are learnt using labeled data while respecting concept relationships. If labeling a concept is much cheaper than labeling a document, and if Eq. 2 can provide an accurate representation of the true underlying distribution, then it is clear that our framework can lead to a much more efficient learning mechanism in comparison to the traditional process. We present strong empirical evidence that supports this statement. We now describe the three main steps of our framework in more detail.

2.1 Documents-to-Ontology Distribution

As part of the specification of the Ontology, we define a feature function ψ_{ont} that extracts a set of attributes for any given concept v, as well as any given document d. The role of ψ_{ont} is to provide a feature space in which the similarity between documents and concepts can be measured. Let $N_k(v)$ denote the k-neighborhood of the concept v i.e., the set of concepts connected to v by a path of length upto k (We used $k = 3$ in our experiments), comprising of directed edges in E. We define the documents-to-ontology distribution as follows,

$$P(v|d) \propto \sum_{q \in N_k(v)} \psi_{ont}(d)^T \psi_{ont}(q) \tag{3}$$

Note that this distribution naturally takes the graph structure of concepts into account. The definition of ψ_{ont} is domain/task independent and essentially specifies a general procedure to match documents against the ontology. Therefore, this step is the unsupervised component of our framework. Note that implicit in the definition above is the assumption that document d is not orthogonal to *all* the concepts $v \in V$, with respect to the feature space induced by ψ_{ont}. This assumption allows similarity scores to be correctly normalized into a probability distribution. Documents that do not satisfy this assumption are considered out of coverage in the model.

2.2 Ontology-to-Class Distribution

The ontology-to-class distribution is estimated from a labeled sample $\{v_i, y_i\}_{i=1}^l$ and is the only component of our system where human supervision is expected. In comparison to reading, comprehending and labeling documents, the rapid identification of concept-class associations can be a much more effortless and time-efficient exercise. The task of labeling graphs from partial node labeling has received significant recent attention in machine learning, with rigorous regularization frameworks to handle both undirected [Belkin *et al.*, 2004] and directed cases [Zhou *et al.*, 2005]. These methods may be seen as smooth diffusions or random-walk based propagation of labeled data along the edges of the graph. In particular, let $\mathbf{p}$ be a vector [2] such that $p_i = P(y = 1|v_i)$. Then one can solve the following optimization problem,

$$\mathbf{p}^\star = \arg\min_{\mathbf{p}} -\frac{1}{l}\sum_{i=1}^{l} \log\left[p_i^{\frac{1+y_i}{2}}(1-p_i)^{\frac{1-y_i}{2}}\right] + \gamma \mathbf{p}^T L \mathbf{p}$$
$$\text{subject to:} \quad 0 \le p_i \le 1, i = 1 \ldots |V|$$

where the first term is negative log-likelihood and the second term measures smoothness of the distribution with respect to the ontology as measured using the Laplacian matrix [Zhou *et al.*, 2005] of the directed graph (V, E) with $\gamma > 0$ as a real-valued regularization parameter.

Another simple and very effective choice used in our experiments is a *"hard" label propagation* where $P(y = 1|v) = 1$

[1] Our methods also generalize to multiclass problems.

[2] The parameters β in Eq. 2 can be identified with $\mathbf{p}$

for all v exclusively in the neighborhood of a positively labeled concept node, $P(y = -1|v) = 1$ for all v exclusively in the neighborhood of a negatively labeled concept node, and $P(y = 1|v) = 0.5$ for the remaining concepts.

2.3 Final Classifier Induction from Unlabeled Data

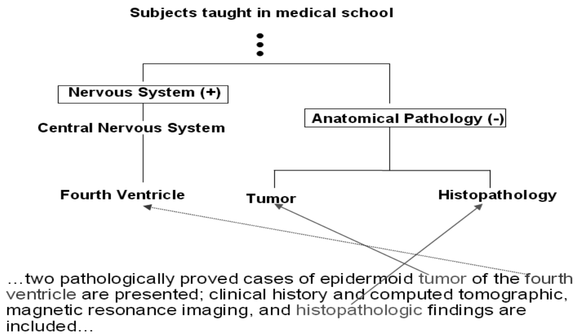
The steps described above allow a documents-to-class distribution 2 to be estimated with low-cost concept-level supervision. We can now define the ontology-based classifier,

$$O(d) = \underset{y \in \{-1,+1\}}{\arg\max}\ P_{ont}(y|d) \tag{4}$$

Note that if $P_{ont}(y = 1|d) = P_{ont}(y = -1|d) = 0.5$, then $O(d)$ is not uniquely defined. This can happen, for example, when $P(v|d) > 0$ implies $P(y = 1|v) = P(v = -1|v)$, i.e, the document d matches concepts where the class distributions are evenly split. Documents for which the distribution in Eq. 3 cannot be properly defined, or for which $O(d)$ is not uniquely defined are considered out of coverage. Let $\mathcal{C}$ be the set of documents that have coverage. We can now take our entire original unlabeled collection, $\{d_i\}_{i=1}^n$ and generate a labeled set $\{(d_i, O(d_i)) : d_i \in \mathcal{C}\}$. In the final step of our framework, we use this labeled set, obtained using concept labeling instead of direct document labeling, to train a classifier via Eq. 1. This is done for the following reasons: (1) this allows generalization to test documents that are not covered by the ontology-based classifier (Eq. 4), and (2) even if the ontology-based classifier only weakly approximates the true underlying Bayes optimal classifier, the labels it generates can induce a strong classifier in the bag-of-words representation. This is because highly domain-specific word dependencies with respect to classes, not represented in ontology-specific attributes, may be picked up during the process of training. We refer to the traditional process as *document labeling* and contrast it with our *concept-labeling* framework. The direct use of Eq. 4 is referred to as *ontology-based classification.*

3 System Overview

We now describe a text categorization system that implements our framework using the English-only subset of Wikipedia. As a directed graph, our Wikipedia Ontology comprises of about 4.1 million nodes with more than 20 million edges. About 85% of the nodes do not have any subcategories and are standalone concepts. Each concept has an associated webpage with a title and a detailed text description. We setup the feature map ψ_{ont} using the vocabulary space of $|V|$ concept titles. For any concept v, we define a binary vector $\psi_{ont}(v)$ which is valued 1 for the title of v and 0 otherwise. For any document d, the vector $\psi_{ont}(d)$ is a "bag-of-titles" frequency vector obtained by indexing d over the space of concept titles. Our indexing is robust to minor phrase variations, i.e., any unigram, bigram or trigram token that redirects to a Wikipedia page is indexed against the title of that page. Then, the documents-to-ontology distribution, Eq. 3, $P(v|d)$, is proportional to the number of occurences of titles in the document for all concepts in the neighborhood of v. This unsupervised step of mapping documents onto the ontology is schematically shown in Figure 1. To specify the ontology-to-class distribution, we allow the user to search Wikipedia or browse the category tree[3] and supply a collection of labeled concepts. We induce the ontology-to-class distribution by identifying entities from the Wikipedia ontology in the documents to be labeled. If we find more entities from the sub-tree corresponding to Class 1 as opposed to Class 2, we label the document as Class 1. If no entities belonging to the Wikipedia sub-tree of either class are found in the document, the document cannot be labeled. This procedure is used to label a large number of labeled data from unlabeled examples, with which we train a multinomial Naive Bayes classifier with respect to bag-of-words representation, as in Eq. 1



...two pathologically proved cases of epidermoid tumor of the fourth ventricle are presented; clinical history and computed tomographic, magnetic resonance imaging, and histopathologic findings are included...

Figure 1: An illustrative example showing the unsupervised mapping of terms in a document to part of an ontology, specifying the documents-to-ontology distribution (Eq. 3). Two concepts have been labeled as + (*nervous system*) and - (*neoplasm*) from which an ontology-to-class distribution is induced. Based on Eq. 4 this document would be labeled as *neoplasm.*

4 Empirical Evaluation

In this section we describe our datasets, followed by experiments and discussion of results.

4.1 Datasets

We evaluated the effectiveness of our methods on a diverse collection of text categorization problems spanning social media content, medical articles and newsgroup messages.

Smartphones: An important application for text classification is filtering social media streams such as blogs and Twitter for relevant content. Human labeling in such scenarios is prohibitive since several such relevance models may need to be rapidly and simultaneously built. With on the order of 100M public tweets in Twitter per day, there is a strong need to provide a filtering capability to allow users to see only tweets relevant to a subject of interest. Many tools, including Twitter itself, offer the capability to search tweets using keywords. However, many broad subjects cannot be exhaustively characterized with a small set of easily identifiable keywords expected to be present in short pieces of text with less than 140 characters. This makes it necessary to build text classification

[3] http://en.wikipedia.org/wiki/Special:CategoryTree

models. Here, we consider the task of identifying tweets discussing smartphones, for which we created a labeled dataset consisting of 420 positive and negative (not *smartphone*) examples. The positive examples were labeled independently by three people. The negative examples were randomly sampled from an archive of over 1M tweets. For this dataset we report results of 10-fold cross-validation.

20 newsgroups: The task here is to classify messages belonging to various newsgroups. We pose several binary classification tasks among closely related [4] newsgroups. We use the standard train-test splits provided with this data.

ohsumed.91: This collection is taken from the Ohsumed corpus [5] which contains MEDLINE documents for the year 1991. The task is to assign documents to one of the four most frequent MeSH disease categories. For our experiments, we removed documents that belonged to more than one category. The collection was then split into equal sized train–test sets.

4.2 Experimental Methodology

The concept labels we used in our experiments are listed in Table 1. For each class we list the Wikipedia categories (concepts) that we associate with it. These categories were assigned by simply searching for the class names in Wikipedia and browsing the related Wikipedia ontology. Note that, most concept labels are fairly obvious, and we assigned at most 4 Wikipedia categories to a class, which requires less than 5 minutes of human supervision per class.

The best performance we can expect on these datasets is using human labels on all available training examples. We report results on using Naive Bayes with human labeled data, which we refer to as *Document Labeling.* For completeness we also report results using SVMs for the same data. We compare Concept Labeling to these benchmark to see how close to Document Labeling we can get. In addition, we compare Concept Labeling to 3 other baselines. For the first baseline, *Wiki Transfer,* we take all the pages in Wikipedia corresponding to the categories listed in Table 1, and use these as training examples with Naive Bayes for each binary classification task. In addition to providing labels for training data, the same approach in Concept Labeling can be used to label test documents (as in Eq. 4). Since not all documents can be mapped to Wikipedia categories that are relevant to our tasks, we would expect that such an approach will leave some test examples unlabeled. However, we can still measure the accuracy on the test examples that can be labeled. We refer to this baseline as *Ontology-only Classification* (OC). Finally, we also compare to an alternative approach to using little supervision via semi-supervised learning. In particular, for each data set we use 100 hand-labeled examples, and build a *Transductive SVM* (TSVM)[Joachims, 1999], treating the remaining examples in the training set as unlabeled.

4.3 Results

All our results are summarized in Table 2. First, we note that training on all hand-labeled examples, Document Labeling, using Naive Bayes or SVMs does not make much difference, with mean accuracies being within 1% of each other. For consistency, we will use Naive Bayes as the benchmark for discussion, since the other systems being compared also use the same base classifier. The results show that on average we are able to achieve 93% of the predictive power we can get from hand-labeling thousands of examples. This is remarkable since we needed to provide only a simple mapping between a few Wikipedia categories and the corresponding classes to achieve this.

Target Class	**Wikipedia Categories**
autos	Automobiles
motorcycles	Motorcycles, Motorcycle technology, Motorcycling
baseball	Baseball
hockey	Ice hockey
guns	Firearms, Weapons, Ammunition, Gun politics
mideast	Middle East, Western Asia, North Africa
pc	IBM PC compatibles, IBM personal computers
mac	Macintosh computers, Macintosh platform
cardiology	Cardiovascular system
immunologic	Immunology
neoplasm	Oncology, Anatomical pathology
nervous system	Nervous system
smartphones	Smartphones

Table 1: Concept labels: Labels for Wikipedia categories

Given that we are providing labels to Wikipedia categories, an obvious alternative to Concept Labeling, is using the documents in these Wikipedia categories directly to induce a classifier. This method, Wiki Transfer, is equivalent to the simple approach to transfer learning, where labeled documents in a source domain are used to train a classifier for the target domain. While this approach actually does outperform Concept Labeling on two data sets, in general it performs quite poorly. This is because the distribution of documents in Wikipedia can be quite different from the distribution in our target domains. So, in order to make such a method effective, more sophisticated approaches to account for the covariance shift [Bickel *et al.*, 2009] would need to be employed.

Ontology-only Classification (OC, Eq. 4), where the document-to-ontology and ontology-to-class distributions are used to directly label test examples, performs better than naive transfer learning with Wiki Transfer. However, this approach suffers from the drawback that it may not be able to label all examples. If a document does not contain terms that can be mapped to the ontology, or if the mapped terms are not relevant to the target classes, then OC is unable to provide a label. In Table 2, the column *Coverage Percentage* lists the percentage of test examples for which OC was able to provide a label, which can be as low as 65%. For labeling a training set, this is not a significant problem, since we can still build a classifier with fewer examples. However, if the application requires that all test instances be labeled, then Ontology-only Classification is not a feasible solution. We also report the ac-

[4] defined by http://people.csail.mit.edu/jrennie/20Newsgroups

[5] http://ir.ohsu.edu/ohsumed/ohsumed.html

Data Set	Document Labeling		Concept Labeling	Wiki Transfer	Coverage Percentage	Ontology-only Classification	TSVM
	NB	SVM					
baseball-hockey	96.61	93.72	96.61	83.04	81.53	78.26	90.70
guns-mideast	98.38	96.89	96.89	74.32	82.70	76.35	94.59
cardiology-immunologic	96.35	96.36	95.96	63.46	91.15	82.50	93.65
immunologic-nervous system	93.13	94.89	92.61	79.92	92.43	83.45	90.32
immunologic-neoplasm	92.53	92.71	91.87	41.11	92.03	88.37	92.71
cardiology-neoplasm	96.79	97.17	91.67	95.40	89.70	81.07	95.74
nervous system-neoplasm	95.52	96.43	87.94	63.09	91.48	79.88	90.99
cardiology-nervous system	84.49	86.53	78.77	58.70	89.60	75.31	73.27
pc-mac	89.32	88.42	77.22	74.77	64.73	55.98	78.38
autos-motorcycles	96.22	95.97	74.06	83.37	79.35	58.06	76.07
smartphones	89.27	96.54	77.84	33.89	100.00	85.93	90.11
Mean Accuracy	93.51	94.44	87.62	68.28	87.15	77.17	87.87

Table 2: Comparing Concept Labeling to other benchmarks based on classification accuracy

curacy of OC on just the examples for which it can provide a label. We see that even on the partial set of examples labeled, the accuracy is not very high compared to the other baselines. However, what is notable is that, on average, training a classifier on a subset (87%) of training examples, which are labeled with moderate accuracy (77%), we are able to build a classifier with higher accuracy (88%) through Concept Labeling. This observation highlights the advantage of inducing a classifier on weakly-labeled training examples, which leads to better generalization perfomance on unseen test examples, over using the noisy labeling process on the same examples.

Further, when compared to Transductive SVMs, Concept Labeling performs comparably on average, with TSVMs producing higher accuracies on 6 of 11 datasets. Recall, that for TSVMs we provided 100 hand-labeled examples, which would require substantially more annotation time than the few minutes it takes to provide the labels in Table 1. Nevertheless, these results confirm that both Concept Labeling and semi-supervised learning are good alternatives to getting the most out of your data and human effort. Furthermore, they are not mutually exclusive. An effective strategy could be to label a few high-confidence training examples by Concept Labeling and then using a semi-supervised approach to leverage the remaining unlabeled examples.

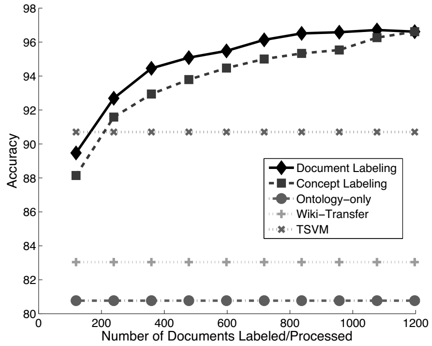


Figure 2: Learning Curve for BASEBALL vs HOCKEY

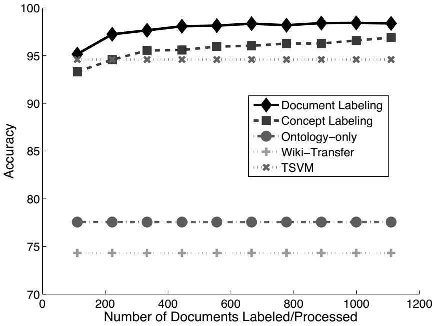


Figure 3: Learning Curve for GUNS vs MIDEAST

The benefits of using Concept Labeling are clearly demonstrated in Figures 2 and 3. Here we present learning curves, with the performance of Document Labeling with increasing amounts of training data. In the case of Concept Labeling, we have a fixed number of concept labels, and the x-axis corresponds to the number of unlabeled documents presented for labeling. Note that, since, not all unlabeled documents are labeled as a result of Concept Labeling, the models built with Concept Labeling are using fewer training documents. The points on this curve demonstrate the improvement in generalization of Concept Labeling with increasing number of documents. However, labels for these additional documents are still based on the fixed initial cost of labeling concepts. The plots highlight the cost-effectiveness of Concept Labeling over Document Labeling. For instance, Fig. 2 shows that with only 2 concept labels can build a classifier that is as accurate as hand-labeling approximately 1200 documents.

Finally, we note that the negative class (*not smartphone*) in the *smartphone* classification task is modeled in a different fashion from other datasets. In principle, everything that excludes smartphones is included in this class. The ontology corresponding to such a class is very huge and is very difficult to model explicitly. For the Wiki Transfer baseline, we randomly picked 1000 documents from Wikipedia as the training documents for this class. We were able to obtain only an accuracy of 34% using this baseline method. For Ontology-only Classification any tweet that does not have a mapping to the smartphone ontology was classified as the negative class. For most applications, OC cannot produce a label for all test examples. However, for *smartphones*, OC has 100% coverage by design. In such cases, OC can be a better alternative than inducing a classifier through Concept Labeling, as demonstrated by the *smartphone* results.

5 Related Works

One of the earliest instances of exploiting external data sources for supervised learning was in using WordNet synonyms and hypernyms [Scott and Matwin, 1998] to build enhanced document representations. In the last few years, the proliferation of collaboratively created, high quality Web 2.0 resources, like Wikipedia, led to several efforts to utilize them for classification [Gabrilovich and Markovitch, 2006; Wang and Domeniconi, 2008]. [Banerjee, 2007] showed that using features in the Wikipedia space makes a classifier more robust in an inductive transfer setting. [Gupta and Ratinovf, 2008] use documents from the Open Directory Project and Yahoo Answers along with Wikipedia to achieve classification accuracies higher than using either one of those resources. In all these cases, the user provides labels on the original document space and significant number of labels are required to achieve good classification accuracy. [Janik and Kochut, 2008] directly classify documents based on Wikipedia categories using a thematic graph construction. Their approach is conceptually similar to our Ontology-only Classification baseline and as we show in Sec. 4, training a classifier is always better to obtain complete coverage and higher accuracy through generalization.

6 Conclusions

In this paper, we propose a novel approach to rapidly building new text categorization models, by shifting human annotation effort from the traditional labeling of documents, to the more cost-effective labeling of concepts in an ontology. We formalized this general framework for Concept Labeling, and presented a specific instantiation using Wikipedia as our ontology, applied to text classification in several domains. Our empirical results show that very little, high-level supervision, in the form of concept labels lead to classifiers that are comparable to using a large number of labeled documents. On average our models can achieve 94% of the accuracy of individually-labeled documents with a very small fraction of the effort. As such, Concept Labeling is a more efficient use of human resources, enabling us to swiftly build classifiers for many new domains. We also demonstrated that our approach produces models that are comparable to exploiting unlabeled examples through semi-supervised learning, and better than using related labeled documents via a naive transfer learning approach. Given that Concept Labeling and semi-supervised learning are complimentary paradigms, exploring their combination is a promising avenue for future work.

References

[Attenberg and Provost, 2010] J. Attenberg and F. Provost. Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *KDD*, 2010.

[Banerjee, 2007] S. Banerjee. Boosting inductive transfer for text classification using Wikipedia. ICMLA, 2007.

[Belkin *et al.*, 2004] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, 2004.

[Bickel *et al.*, 2009] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning under covariate shift. *JMLR*, 10:2137–2155, December 2009.

[Blitzer *et al.*, 2007] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[Chapelle *et al.*, 2005] O. Chapelle, B. Schoelkopf, and A. Zien. *Semi-supervised Learning*. MIT Press, Cambridge, Massachusetts, 2005.

[Druck *et al.*, 2008] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[Ferragina and Scaiella, 2010] P. Ferragina and U. Scaiella. Tagme: on-the-fly annotation of short text fragments (by Wikipedia entities). In *CIKM*, 2010.

[Gabrilovich and Markovitch, 2006] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. In *AAAI*, 2006.

[Gupta and Ratinovf, 2008] R. Gupta and L. Ratinovf. Text categorization with knowledge transfer from heterogeneous data sources. In *AAAI*, 2008.

[Janik and Kochut, 2008] M. Janik and K. J. Kochut. Wikipedia in action: Ontological knowledge in text categorization. *Intl. Conf. on Semantic Computing*, 2008.

[Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, June 1999.

[Melville *et al.*, 2009] P. Melville, V. Sindhwani, and R. Lawrence. Social media analytics: Channeling the power of the blogosphere for marketing insight. In *Proc. of the Workshop on Information in Networks*, 2009.

[Scott and Matwin, 1998] S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *Workshop on usage of WordNet in NLP Systems (COLING-ACL '98)*, pages 45–51, August 1998.

[Settles, 2009] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Wang and Domeniconi, 2008] P. Wang and C. Domeniconi. Building semantic kernels for text classification using Wikipedia. KDD, pages 713–721, 2008.

[Zhou *et al.*, 2005] D. Zhou, J. Huang, and B. Schoelkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML*, 2005.