Topic labeled text classification

Conference Paper · July 2014

DOI: 10.1145/2600428.2609565

Swapnil Hingmire
Tata Consultancy Services Limited
23 PUBLICATIONS 112 CITATIONS

Topic Labeled Text Classification: A Weakly Supervised Approach

Swapnil Hingmire[1,2]
swapnil.hingmire@tcs.com

[1]Systems Research Lab
Tata Research Development and Design Center
Pune, India

Sutanu Chakraborti[2]
sutanuc@cse.iitm.ac.in

[2]Department of Computer Science and Engineering
Indian Institute of Technology Madras
Chennai, India

ABSTRACT

Supervised text classifiers require extensive human expertise and labeling efforts. In this paper, we propose a weakly supervised text classification algorithm based on the labeling of Latent Dirichlet Allocation (LDA) topics. Our algorithm is based on the generative property of LDA. In our algorithm, we ask an annotator to assign one or more class labels to each topic, based on its most probable words. We classify a document based on its posterior topic proportions and the class labels of the topics. We also enhance our approach by incorporating domain knowledge in the form of labeled words. We evaluate our approach on four real world text classification datasets. The results show that our approach is more accurate in comparison to semi-supervised techniques from previous work. A central contribution of this work is an approach that delivers effectiveness comparable to the state-of-the-art supervised techniques in hard-to-classify domains, with very low overheads in terms of manual knowledge engineering.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Experimentation, Performance, Theory, Verification

Keywords

text classification, topic modelling, weakly supervised, semi supervised

1. INTRODUCTION

Text classification helps in organizing and using the knowledge hidden in a large collection of documents such as the World Wide Web. The effectiveness of supervised text classification depends critically on the number and nature of labeled documents. Unfortunately, labeling a large number of documents is a labor-intensive and time consuming process that involves significant human intervention. It is therefore important from a practical standpoint to explore text classification approaches that reduce the time, effort and cost involved in creating labeled corpora. The goal is to make best use of human expertise that is available and bring down cognitive load of labellers.

One potential solution to reduce label acquisition overhead in text classification is to allow annotators to incorporate their domain knowledge into learning. For example, instead of labeling a set of documents, annotators are required to label a set features. These labeled features are then used to guide the learning algorithm. Liu et al. [17] and Druck et al. [8] are a few example approaches in this direction. However, it is not always practical for human annotators to correctly label a set of features without assessing the context. Also the impact of a label on predictions of a classifier are hard to assess, since human labellers have often no access to statistical properties of the collection. The classification performance of such algorithms is sensitive to the feature set chosen for labeling. Hence, there is a need to explore new approaches that will enable annotators to represent their domain knowledge and incorporate it into learning in an efficient manner.

In this paper, we propose a Latent Dirichlet Allocation (LDA)[4] based weakly supervised text classification algorithm, which we call *topic labeled classification* or TLC. Our algorithm is based on the generative property of LDA. LDA is an unsupervised statistical machine learning technique that can be used to uncover the underlying semantic structure of a document collection. LDA automatically finds a likely set of topics over a large document collection and represents each document in the collection in the form of topic proportions. In LDA, a topic can be interpreted by its most probable words. As the topics are interpretable, and we know how a document exhibits each topic, we can infer the *gist* of the document [11].

In our algorithm, we ask an annotator to assign one or more class labels to LDA topics which are short, interpretable and preserve statistical relationships induced from a corpora. We use these labeled topics to classify a document based on its posterior topic proportions. We show how this approach can be augmented by incorporating class labels assigned to few words in the collection.

The paper is organized as follows: In Section 2, we review prior work on semi supervised or weakly supervised text classification methods. Section 3 proposes the central

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.
Copyright 2014 ACM 978-1-4503-2257-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2600428.2609565.

idea behind labeling topics for building a text classifier. In Section 4, we enhance our text classification algorithm by incorporating domain knowledge in the form of labeled words. Section 5 demonstrates effectiveness of our algorithm with experiments on four datasets. We conclude our paper with future prospects of our work in Section 6.

2. BACKGROUND AND RELATED WORK

In semi-supervised learning [7], supervision is provided by labeling a small number of documents. These labeled documents and a large number of unlabeled documents are used while learning the classifier. While estimating the parameters of the classifier, certain assumptions about the distribution of labeled and unlabeled documents will have to hold. Common assumptions are the cluster assumption [23], the low-density separation assumption [9], the manifold assumption [31], the transduction assumption [13], and the multi-view assumption [5]. These semi-supervised learning methods are based on very strong assumptions about the relationship between class labels and the distribution of both labeled and unlabeled data, so if the data does not satisfy assumptions necessary for methods discussed above, it is unrealistic to achieve desired classification performance. These methods are also sensitive to the set of initially labeled documents and hyper-parameters used while learning the classifier.

Several researchers have proposed text classification algorithms that use labels on words (features). Liu et al. [17] propose a text classification algorithm by labeling words. At the beginning of this algorithm unlabeled documents are clustered. Then, feature selection on the resulting clusters is performed to rank words according to their discriminative power. Then, experts analyze the ranked list and for each class select a small set of representative words. Eventually, these small sets of representative words are used to create a text classifier using the combination of naive Bayes classifier and the Expectation-Maximization (EM) algorithm.

Druck et al. [8] propose a generalized expectation (GE) criteria based maximum entropy text classifier using labeled words (GE-FL). As the performance of many machine learning based text classifiers is sensitive to model parameters and features, it is important as well as challenging to find optimal model parameters and high quality features. Instead of injecting domain knowledge through the selection of model parameters and feature selection, GE provides a way for humans to directly express preferences to the parameter estimator naturally and easily using the language of "expectations" [19]. For example, in the *baseball-hockey* text classification problem, GE-FL allows humans to specify preferences; for example, the presence of the word "puck" strongly represents the class *hockey*. In GE-FL these preferences can be readily translated into constraints on model expectations.

However, given an unlabeled dataset, it is challenging to come up with a small set of words that should be presented to the annotators for labeling. As text classifiers use various statistical techniques, each word should be statistically predictive of some class. At the same time, it is important to ensure that the words are not only statistically discriminative but also meaningful to humans who are labeling them. For example, in the *pc-mac* subset of the 20Newsgroup dataset[1] the word "isa" statistically predicts the class *pc* because 97% of training documents containing the word "isa" are labeled with the class *pc*, but it is difficult for a human to understand the meaning of this word and label it as either *pc* or *mac*, without knowing its context. Also a human annotator labels individual words one by one, she might not be aware of the context of a word and hence she may discard or mislabel a polysemous word, which may affect the performance of a text classifier.

ID	**Most prominent words in the topic**	**Class (pc / mac)**
0	port modem board advance cable mouse serial irq switch computer	**?**
1	bit ram **mac** simms cards **ibm** **windows** system **nubus** bus	**pc + mac**
2	**controller** ide **dos** rom **bios** isa **tape** sound interface	**pc**
3	**apple** problem **mac** buy price computer power **duo** **powerbook**	**mac**

Table 1: Topic labeling on the *pc-mac* subset of the 20Newsgroup dataset

Blei et al. [4] use LDA topics as features in supervised text classification. As LDA models only words in documents and not their class labels, Blei and McAuliffe [3] propose sLDA, a supervised extension to LDA. DiscLDA [15] and MedLDA [30] are a few supervised extensions of LDA which can be used for text classification, but they need expensive labeled documents.

Hingmire et al. [12] propose a text classification algorithm, ClassifyLDA, based on labeling of LDA topics. In ClassifyLDA algorithm, an annotator assigns a single class label to each topic. However, topics identified by LDA are not always in accordance with the underlying class structure of the corpus, it is possible that a topic represents multiple classes or no class at all. Table 1 represents 4 topics inferred on the *pc-mac* dataset. We can observe that the most prominent words in the topic 0 do not represent a class and in topic 1, they represent both *pc* and *mac* classes. In this algorithm mislabeling of such topics may affect the performance of the classifier.

The basic idea in our algorithm is that an annotator assigns a class label l to a topic t, if its most probable words semantically represent the class l. LDA gives per-document per-word position topic assignments. If the word at nth position in document d is assigned to just one topic t (a word can belong to more than one topic) and the annotator has assigned a class label l to the topic t then, we assign the class label l to the same word position.

We use these per-document per-word position label assignments to create a new LDA model and update its parameters using collapsed Gibbs sampling [10] by assuming asymmetric Dirichlet prior over topic distribution of each document [28]. As we want to find probability of generating a document by each class label, in the new LDA model we set the number of topics to the number of class labels, such that a topic represents a class. Given a test document, we infer its probability distribution over the class labels using the new LDA model and classify it to its most probable class label. We deal with the topics representing multiple classes using Topic-in-Set knowledge [1] mechanism. This mechanism allows us to add partial supervision to LDA to encourage it to recover topics relevant to user interest. We

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

also enhance our text classification algorithm by incorporating domain knowledge in the form of labeled words by simply updating the asymmetric prior.

In our algorithm we assume asymmetric Dirichlet prior over topic distribution of each document, because in LDA topics, it is possible that same word appears with high probability in the topics with different class labels, which is likely to be a dataset specific stop word. In case of symmetric prior the topic assignments to these words may span over too many topics which may lead to incorrect classification.

In asymmetric prior structure, the probability of the word at nth position in a document being assigned to topic t depends on the number of topic assignments that previously matched the topic t in the same document [28]. So in asymmetric prior structure, the topic assignments to these words are restricted to a very few topics and it can play an important role when the classes are not clearly separable.

LDA uses word co-occurrence to find topics, so the most probable words of a topic are likely to be semantically related to each other. These most probable words are representative of the dataset, so there is no need for the annotator to search for the right set of features for each class. Also, the annotator can resolve the sense of a polysemous word by observing other most probable words in a topic.

In our algorithm, an annotator does not read and label individual documents. Instead, she labels interpretable topics. These topics are inferred in an unsupervised manner, also they are very few, when compared to the number of documents. Labeling LDA topics overcomes several issues of labeling individual words such as finding a small set of both statistically and semantically predictive words to be labeled. We deal with the topics representing multiple classes using Topic-in-Set knowledge mechanism. We also present an extension of the TLC approach where the annotator, in addition to labeling topics, also labels a few words. Our approach however does not need a labeled document collection for identifying discriminative words. Interestingly, our experiments reveal that significant improvements can be obtained when the words presented to the annotator are identified using document labels inferred using TLC. Additionally, if the annotator explicitly labels a few words, incorporating these labeled words significantly improves text classification performance on harder datasets like *pc-mac*.

3. TOPIC LABELED CLASSIFICATION

In this section, we propose our text classification algorithm: TLC, based on labeling of LDA topics.

3.1 LDA Model

LDA is a probabilistic *generative* model. It describes the following procedure for generating documents using simple probabilistic rules. (Refer Table 2 for the notations used in this paper.)

1. Select the word probabilities for each topic t:
   $\phi_t \sim \text{Dirichlet}(\beta_w)$
2. Select the topic proportions for the document d:
   $\vartheta_d \sim \text{Dirichlet}(\alpha_t)$
3. Select the topic for each word position:
   $z_{d,n} \sim \text{Multinomial}(\vartheta_d)$
4. Select each word:
   $w_{d,n} \sim \text{Multinomial}(z_{d,n})$

3.2 Training of TLC

Let D be a set of unlabeled training documents and the task is to build a text classifier, which will classify a given document into one of the K classes in C.

We use collapsed Gibbs sampling algorithm [10] to learn T number of topics on D. The number of topics T is specified by a human annotator and it is typically greater than K. Let Z be the hidden topic structure (i.e. per document per word topic assignment) of the LDA model M learned on D.

We represent a topic by ordering the words in the vocabulary of D in the decreasing order of their probability of observing under the topic. We ask the human annotator to assign a class label $c_i \in C$ to a topic t based on its most probable words. Let, *L(t)* be the function which returns the class label assigned to the topic t by the human annotator.

Now, we will learn a new LDA model M' based on the labeled topics. We name this topic labeled LDA model as TL-LDA model. We use the TL-LDA model to find probability of generating a document by a class on the basis of labeled topics, so in this model we set the number of topics same as the number of classes (K), such that a topic represents a class. Let, Z' be the hidden topic structure for the corpus D for the new model M'. We initialize $z'_{d,n}$ i.e. the topic assigned to the word $w \in W$ at the position n in document d as follows:

$$\text{If } z_{d,n} = t \text{ and } L(t) = c \text{ then } z'_{d,n} = c \tag{1}$$

We then update M' using collapsed Gibbs sampling. The Gibbs sampling equation used to update the assignment of a class label $c \in C$ to the word $w \in W$ at the position n in document d, conditioned on α'_d, β_w is:

$$P(z'_{d,n} = c | z_{d,\neg n}, w_{d,n} = w, \alpha'_d, \beta_w) \propto \frac{\psi'_{w,c} + \beta_w - 1}{\sum_{v \in W} \psi'_{v,c} + \beta_v - 1} \times (\Omega'_{c,d} + \alpha'_{d,c} - 1) \tag{2}$$

We use a subscript $d, \neg n$ to denote the current token, $z_{d,n}$ is ignored in the Gibbs sampling update. α'_d is a K-dimensional vector and asymmetric Dirichlet prior over topics for the document d. This prior is similar to the asymmetric Dirichlet prior used in Prior-LDA [24]. It is computed as follows:

$$\alpha'_d = \left[\eta * \frac{N_{d,1}}{I_d} + \alpha, \eta * \frac{N_{d,2}}{I_d} + \alpha, ..., \eta * \frac{N_{d,K}}{I_d} + \alpha\right] \tag{3}$$

where I_d is the current number of words drawn from ϑ'_d for document d and $N_{d,c}$ is the current number of topic assignments in the document d matched to the topic c. α'_d is a scaled, smoothed, normalized vector of topic assignments [24]. The hyper-parameter η specifies the total weight contributed by the observed topic assignments and the hyper-parameter α is an additional smoothing parameter for each topic.

After performing collapsed Gibbs sampling using equation 2, we use Z' to compute a point estimate of the distribution over words $\phi'_{w,c}$ and a point estimate of the posterior distribution over class labels for each document d (ϑ'_d):

$$\phi'_{w,c} = \frac{\psi'_{w,c} + \beta_w}{\left[\sum_{v \in W} \psi'_{v,c} + \beta_v\right]} \tag{4}$$

$$\vartheta'_{c,d} = \frac{\Omega'_{c,d} + \alpha'_{d,c}}{\left[\sum_{i=1}^{C} \Omega'_{i,d} + \alpha'_{d,i}\right]} \tag{5}$$

D = $\{d\}$	The set of all documents in the training corpus
C = $\{c_1, c_2, ..., c_K\}$	The set of class labels of the corpus D
N_d	The number of words in document d
y_d	The class label of document d
W = $\{w\}$	The set of all unique words (vocabulary) in the corpus
T	The number of topics specified as a parameter
Z = $\{z_{d,n}\}_{d=1:\lvert D\rvert, n=1:N_d}$	Per document per word position topic assignment
α_t	The parameters of topic Dirichlet prior for the topic t
β_w	The parameters of word Dirichlet prior for the word w
$w_{d,n}$	The word at position n in the document d
$z_{d,n}$	The topic assigned to the word at position n in the document d
$\psi_{w,t}$	The count of word w assigned to topic t
$\Omega_{t,d}$	The count of the topic t assigned to some word w in document d
ϕ_t	The word probabilities for topic t
$\phi_{w,t}$	The probability of the word w observed under the topic t
ϑ_d	The topic probability distribution for document d
$\vartheta_{t,d}$	The probability of generating document d by topic t
$\Phi = \{\phi_t\}$	The topic-word distributions for all words in W over all topics Z
$\Theta = \{\vartheta_d\}$	The document-topic distributions for all the documents in D

Table 2: Key notations

3.3 Inference

Inference for a test document d using M' involves estimating its distribution (ϑ'_d) over class labels, based on the words in it. ϑ'_d is estimated by iteratively updating the topic assignments of words in the document d to class labels through the Gibbs sampling update:

$$P(z_{d,n} = c|z_{d,\neg n}, w_{d,n} = w, \alpha'_d, \beta_w, \phi'_{w,c}) \propto \phi'_{w,c} \times (\Omega'_{c,d} + \alpha'_{d,c} - 1) \quad (6)$$

where $\phi'_{w,c}$ is estimated while training the M' model and α'_d can be computed using equation 3. After performing certain iterations of Gibbs update, ϑ'_d can be computed using equation 5. Let i be the class label which generated document d with highest probability. So the class label y_d for the document d will be i: $y_d = \underset{i}{\text{argmax}}\, \vartheta'_{i,d}$

3.4 Topics with Multiple Class Labels

Table 1 shows an example where a topic represents multiple classes or no class at all. We use Topic-in-Set knowledge technique proposed in [1] to deal with such a topics.

If a topic represents more than one classes then we allow the annotator to assign multiple class labels to the topic. If a topic does not represent any class at all, then the annotator will assign all the class labels to it. Now, we change the *L(t)* function to return a set of class labels assigned to the topic t. If $z_{d,n} = t$ and $|L(t)| \geq 1$ then, we choose a class $c_j \in L(t)$ randomly and assign it to $z'_{d,n}$. Then we sample a class label for $z'_{d,n}$ using Gibbs sampling, restricted to class labels in *L(t)* only. Let,

$$q_{w,c} = \frac{\psi'_{w,c} + \beta_w - 1}{\sum_{v \in W} \psi'_{v,c} + \beta_v - 1} \times (\Omega'_{c,d} + \alpha'_{d,c} - 1) \quad (7)$$

We modify the Gibbs update equation as follows:

$$P(z'_{d,n} = c|z_{d,\neg n}, w_{d,n} = w, \alpha'_d, \beta_w) \propto q_{w,c} \times \mathbb{I}(c, L(t)) \quad (8)$$

where $\mathbb{I}(c, L(t)) = 1$ if $c \in L(t)$ otherwise 0.

We repeat this process for a number of iterations to get an approximate initial value for $z'_{d,n}$. Then we update the model M' using Gibbs sampling as discussed earlier. Our text classification algorithm is summarized in the Table 3.

1. **Input:** $D = \{d\}$: Unlabeled document corpus and T : number of topics.
2. **Classification of topics:**
   - (a) Learn T number of topics on D for LDA model using collapsed Gibbs sampling. Let $M =< \Phi, \Theta, Z >$ be the hidden topic structure.
   - (b) Ask an annotator to visualize a topic and assign one or more class labels from set C to it.
3. **Initialization of new model M' based on the labeled topics:**
   - (a) If $z_{d,n} = t$ then $z'_{d,n} = c$ such that $c \in L(t)$.
   - (b) Repeat for a number of iterations
     For each document $d \in D$ and for each word position $n = 1$ to N_d,
     - i. if $z'_{d,n} = c$ and $w_{d,n} = v$ then,
       Sample a new class label c' from the distribution in equation 8 s.t. $z'_{d,n} = c'$
4. **Updating M'**
   - (a) Update M' using collapsed Gibbs sampling using equations 2 and 3
5. **Classification of an unlabled document d**
   - (a) Infer ϑ'_d for document d using M'_D.
   - (b) $y_d = \underset{i}{\text{argmax}}\, \vartheta'_{i,d}$

Table 3: Topic labeled text classification algorithm (TLC)

4. INCORPORATING LABELED WORDS

In this section we study how to incorporate domain knowledge in the form of labeled words in the TLC algorithm discussed in previous section.

Labeled words provide affinities between words and classes [8]. If a word w_i is labeled with *jth* class, then we assume that, the word w_i in a document d increases the probability of generating the document d by *jth* class. Hence, the probability of generating a document by *jth* class is proportional to the number of words in the document labeled with *jth* class. We incorporate the labeled words in TLC by modifying the document specific asymmetric prior (equation 3) in the Gibbs sampling update as follows:

$$\begin{aligned}\alpha'_d \quad = \quad & [\eta * \frac{N_{d,1} * m_{d,1}}{I_d} + \alpha, \eta * \frac{N_{d,2} * m_{d,2}}{I_d} \\ & +\alpha \quad , ..., \eta * \frac{N_{d,K} * m_{d,K}}{I_d} + \alpha] \end{aligned} \tag{9}$$

where, $m_{d,j}$= 1 + the number of words in the document d labeled with *jth* class.

As the labeled words are discriminative, incorporating them will encourage the model to identify topics that are aligned to the class level structure of the documents and hence they will improve performance of the classifier. We name this enhancement of TLC algorithm as TLC++.

4.1 Labeling Words

We can obtain word labels from an annotator or using Information Gain technique based on labeled dataset [29]. In the Information Gain based technique, we first rank words in the vocabulary by their Information Gain. To compute Information Gain, we can use the oracle labeler who will reveal the true class labels of the unlabeled documents. However, this would entail high knowledge engineering overhead which is antithetical to the central philosophy of this paper. We explore an alternative approach that labels the unlabeled documents using TLC algorithm in Table 3 and computes Information Gain for each word in the vocabulary of D.

We take the top P words from the vocabulary based on their Information Gain and assign a class label to each word with which it occurs most frequently. Table 4 describes procedure to enhance the TLC algorithm by incorporating labeled words.

1. **Input:** $D = \{d\}$: Unlabeled document corpus and T : number of topics.
2. Classify all the documents in D using TLC algorithm discussed in Table 3 and assume these auxiliary labels as their true labels.
3. Select top P words using Information Gain based on the auxiliary labels of all the documents in D.
4. Label each top word with the class with which it occurs most frequently.
5. Build a new TL-LDA model based on the topics labeled in step 2 and update it to incorporate the labeled words using the Gibbs sampling update in the Equations 8 and 9. We name this enhancement of TLC algorithm as TLC++.
6. Classify test documents using TLC++, based on their most probable topic.

Table 4: Enhancing TLC by incorporating labeled words (TLC++).

5. EXPERIMENTAL EVALUATION

We determine the effectiveness of our algorithm in relation to two weakly supervised text classification algorithms: GE-FL [8] and ClassifyLDA [12]. We evaluate and compare our text classification algorithm by computing Macro averaged F1. Macro-F1 ensures that large classes do not dominate smaller ones; however micro-averaged results are a measure of effectiveness on the large classes in a test collection [18] (e.g. in "wheat-rest" dataset of Reuters-21578, there are total 2691 test documents out of which only 71 documents are labeled with the class "wheat"). So to get a sense of effectiveness on small classes we report Macro-F1. As the inference of LDA is approximate, we repeat all the experiments for each dataset ten times and report average Macro-F1.

We compare TLC with ClassifyLDA by using the same set of labeled topics so that the labeling efforts for both the algorithms are of the order of the number of topics (T). We compare TLC++ with GE-FL using TLC labeled words, this ensures that for both the algorithms, the labeling efforts are of the order T. Though TLC++ does not require any labeled documents, in order to do similar experiments as in [8] and for fair comparison with GE-FL, we assume there exists an oracle who can reveal the labels of unlabeled documents to select most predictive words. We compare TLC++ with GE-FL using the oracle labeled words, so the labeling efforts for TLC++ are of the order $T + |D|$ and for GE-FL, of the order $|D|$. As the number of topics are very few as compared to the number of documents, both algorithms need almost similar labeling efforts. It may be noted that while this comparison is done for the sake of completeness, the real strength of TLC++ is its effectiveness in the setting where words are labeled automatically using feature selection over TLC labeled corpus, and not when oracle labels are used.

5.1 Datasets

We use the following datasets in our experiments.

1. **20 Newsgroups (20NG):** This dataset contains messages across twenty different UseNet discussion groups. These twenty newsgroups are grouped into 6 major clusters. We use different subsets of this dataset for our experiments. The messages in this dataset are posted over a period of time. We use the *bydate* version of the 20Newsgroups dataset [2]. This version of the 20Newsgroups dataset contains 18,846 messages and it is sorted by the date of posting of the messages. This dataset is divided into training (60%) and test (40%) datasets. We construct classifiers on training datasets and evaluate them on test datasets. Table 5 gives details of subsets of this dataset. $|D_{train}|$ and $|D_{test}|$ denotes number of training documents and test documents in a subset respectively. *Avg.* N_d denotes average length of documents in a subset.

2. **SRAA:** This is a UseNet dataset[3] for text classification that describes documents in Simulated/Real/Aviation/Auto classes. This dataset contains 73,128 UseNet articles from four discussion groups for simulated auto racing (sim auto), simulated aviation (sim aviation), real autos (real auto), real aviation (real aviation). This dataset can be viewed in following three different ways depending on the user's need.

a. sim auto vs sim aviation vs real auto vs real aviation

b. auto (sim auto + real auto) vs aviation (sim aviation + real aviation)

c. simulated (sim auto + sim aviation) vs real (real auto + real aviation)

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

[3] `http://people.cs.umass.edu/~mccallum/data.html`

Dataset	$\lvert D_{train} \rvert$	Avg. N_d (training)	$\lvert D_{test} \rvert$	Avg. N_d (test)
med	594	116.77	396	111.63
space	593	125.90	394	110.29
pc	590	73.70	392	66.24
mac	578	64.77	385	64.08
autos	594	79.05	396	73.04
motorcycles	598	73.10	398	65.57
baseball	597	76.80	397	85.28
hockey	600	113.96	399	92.99
politics	2936	91.87	1955	93.81
religion	1456	122.86	968	134.38
comp	2936	91.87	1955	93.81
sci	2373	116.83	1579	96.13
rec	2389	85.76	1590	79.23

Table 5: Details of 20Newsgroups dataset

We randomly split SRAA dataset such that 80% is used as training and 20% is used as test data. Table 6 gives details of subsets of this dataset.

Dataset	$\lvert D_{train} \rvert$	Avg. N_d (training)	$\lvert D_{test} \rvert$	Avg. N_d (test)
realauto	3836	62.49	960	58.75
realaviation	13956	68.00	3489	67.26
simauto	33080	58.12	8271	58.61
simaviation	7700	58.03	1926	60.64
real	17792	66.81	4449	65.42
sim	40780	58.10	10197	58.99
auto	36916	58.57	9231	58.62
aviation	21656	64.46	5415	64.91

Table 6: Details of SRAA dataset

3. **Reuters-21578:** The Reuters-21578 Distribution 1.0 dataset[4] consists of 12902 articles and 90 topic categories from the Reuters newswire. We use the standard 'ModApte' train/test split. It divides the articles by the date of posting of messages. In this dataset, the later 3299 documents are used for testing, and the earlier 9603 are used for training. Table 7 gives details of subsets of this dataset. $\%|D_{train,c}|$ and $\%|D_{test,c}|$ denote percentage of training and test documents labeled with class c respectively.

Dataset (c)	$\%\lvert D_{train,c} \rvert$	Avg. N_d	$\%\lvert D_{test,c} \rvert$	Avg. N_d
acq	23.08	65.92	25.90	62.87
corn	2.61	97.50	2.07	102.45
crude	5.47	116.61	6.90	97.48
earn	40.16	45.23	39.07	37.94
grain	6.40	93.36	5.64	85.87
interest	4.93	83.02	4.78	95.22
money-fx	7.67	96.14	6.54	99.34
ship	2.77	93.34	3.25	92.75
trade	5.17	129.01	4.22	128.68
wheat	3.07	92.78	2.64	79.73

Table 7: Details of Reuters-21578 dataset

4. **WebKB:** The WebKB dataset[5] contains contains 8145 web pages gathered from university computer science departments. The task is to classify the webpages as *student, course, faculty* or *project.* We randomly split this dataset such that 80% is used as training and 20% is used as test data. Table 8 gives details of subsets of this dataset.

Dataset	$\lvert D_{train} \rvert$	Avg. N_d (training)	$\lvert D_{test} \rvert$	Avg. N_d (test)
student	1310	103.07	331	102.77
faculty	896	193.82	228	140.96
course	742	173.29	188	195.11
project	402	183.97	102	215.05

Table 8: Details of WebKB dataset

All datasets were processed using lowercased unigram words, with HTML tags, stop-words and words with length less than three removed.

5.2 Experimental Settings

For various subsets of the datasets discussed above, we choose number of topics as twice the number of classes. In the case of SRAA dataset we inferred 8 topics on the training data and labeled these 8 topics for all the three classification tasks discussed above. As the articles in Reuters-21578 dataset belong to multiple categories, we build binary classifier for each of the ten most frequent classes to identify the news topic as in [23]. We inferred 20 topics on the training data and labeled these 20 topics. We did all the 10 binary classification tasks on these 20 topics. While labeling a topic, we show its 30 most probable words to the annotator.

Similar to [27], we assume symmetric Dirichlet word prior (β_w) for each topic and we set it to 0.01. Similar to [24], in the asymmetric prior over document topic distribution (equation 3), we set $\eta = 50$, the hyperparameter $\alpha = 0.0$ at the time of training and $\alpha = 0.01$ at the time of testing.

We use the implementation of GE-FL from the 2.0.7 version of the MALLET toolkit[6]. For each dataset, we experiment with GE-FL using both TLC labeled words as well as the oracle labeled words by reveling the labels of training documents as in [8]. For both types of experiments, we select the top 10 most informative words per class. We use the same labeled words for both TLC++ and GE-FL for fair comparison.

5.3 Results

Table 9 shows experimental results. We can observe that TLC performs better than ClassifyLDA in 20 of the total 21 subsets with the same labeling efforts. Hence, asymmetric prior over document-topic distribution improves the performance of the classifier when compared to that obtained using symmetric prior.

We can also observe that TLC++ performs better than GE-FL in 20 subsets when the words are labeled by TLC algorithm. Interestingly, when we assume an upper bound on feature selection by assuming existence of an oracle, TLC++ performs better than GE-FL in 18 subsets using the oracle labeled words. However, TLC++ algorithm does not perform as expected on the *earn* and *interest* classes of the Reuters dataset and the WebKB dataset.

In the Reuters dataset, classes like *wheat, corn* have very specific definitions. If a document contains the word "wheat" then there is high probability of belonging it to the *wheat* class. This is not true for the *earn* class, it can not be defined by a single word. Bekkerman et al. [2] analyzed this dataset

[4] `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`

[5] `http://www.cs.cmu.edu/~webkb/`

[6] `http://mallet.cs.umass.edu`

						TLC Labeled Words			Oracle Labeled Words	
Labeling efforts→			**TLC (A1)** T	**Classify LDA (A2)** T		**TLC++ (A3)** T	**GE-FL (A4)** T		**TLC++ (A5)** $T + \|D\|$	**GE-FL (A6)** $\|D\|$
Dataset ↓	***T***									
20Newsgroups										
med-space	4		**0.943***	0.926		**0.948***	0.918		**0.947***	0.939
pc-mac	4		**0.680***	0.641		**0.692***	0.492		**0.732***	0.666
autos-motorcycles-baseball-hockey	8		**0.873**	0.854		**0.884***	0.771		**0.931***	0.904
politics-religion	4		**0.922***	0.892		**0.924***	0.836		**0.929***	0.765
politics-sci	4		**0.911***	0.899		**0.918***	0.636		**0.926***	0.618
comp-religion-sci	6		**0.871***	0.836		**0.872***	0.779		**0.884***	0.773
politics-rec-religion-sci	8		**0.858***	0.815		**0.859***	0.702		**0.892***	0.735
SRAA										
realauto-realaviation-simauto-simaviation	8		**0.806***	0.627		**0.834***	0.760		**0.834***	0.798
real-sim			**0.921***	0.879		**0.921***	0.821		**0.922***	0.832
auto-aviation			**0.930***	0.863		**0.930***	0.869		**0.930***	0.867
Reuters-21578										
acq	20		**0.949***	0.687		**0.946***	0.873		**0.954***	0.821
corn			**0.771***	0.679		**0.771***	0.572		**0.747***	0.694
crude			**0.880***	0.703		**0.880***	0.756		**0.878***	0.584
earn			**0.862***	0.789		0.860	**0.862**		0.861	**0.906**
grain			**0.910***	0.716		**0.903***	0.674		**0.905***	0.834
interest			**0.785***	0.698		**0.780***	0.671		0.799	**0.820**
money-fx			**0.805***	0.707		**0.800***	0.630		**0.812***	0.790
ship			**0.769***	0.687		**0.778***	0.601		**0.784***	0.588
trade			**0.747***	0.694		**0.743***	0.536		**0.741***	0.712
wheat			**0.801***	0.684		**0.796***	0.573		**0.811***	0.791
WebKB										
WebKB	8		**0.684***	0.603		**0.669***	0.642		0.743	**0.794**

Table 9: Experimental results (Macro-F1) of text classification on various datasets. A * indicates that algorithms A1/A3/A5 perform significantly better than A2/A4/A6 respectively, using a two-tailed paired t-test, $p = 0.05$.

in detail. They observed that, the word "vs" appears in 87% of the articles of the class *earn* (i.e., in 914 articles among total 1044 of this class). This word appears in only 15 non-earn articles in the test set and therefore "vs" can, by itself, classify *earn* with very high precision. However the word "vs" does not predict the class *earn* semantically.

Similarly, in the WebKB dataset, for the *student* class the words "uci","espn","nba" are the top three most informative words. Table 10 shows top ten most informative words for each class in the WebKB dataset, selected using Information Gain. These words do not predict the class *student* semantically.

We make here an observation that, a word which is statistically predictive of a class is not necessarily semantically predictive of the same class. However, in real life situations humans are likely to label semantically predictive words only. As the performance of the text classification algorithms based on the labeled words highly depends on the statistically predictive words, these algorithms may not perform as per the expectations. We observed that there are situations, however, where labels given by humans can effectively complement ones derived statistically. This is discussed in the following subsection.

Class	**Most informative words of the class**
student	uci espn nba candidate tsinghua advisors uvic surfing listening love
faculty	ucdavis uml editorships annals umiacs committees editorial presidential served chairman
course	quizzes grader cla attendance thur vectra progresses questionnaire mlh enroll
project	investigators desirable throughput achieved sfu personnel sponsors ongoing researchers affiliated

Table 10: Most informative words for each class in the WebKB dataset, selected using Information Gain feature selection technique

5.4 TLC++ with Human Labeled Words

In the *pc-mac* dataset, large number of features are shared by both the classes and hence there is very little separability between the classes [6]. In order to study this dataset and improve classification performance, instead of labeling words automatically we asked a human annotator to label a set of words from topics inferred on the *pc-mac* dataset. The annotator labeled a few words (as shown in Table 11), that are present in top 30 most probable words of inferred topics.

Class Label	Labeled Words per Class
pc	ibm bios dos controller windows
mac	apple mac centris powerbook nubus macintosh quadra

Table 11: Word labeling on the *pc-mac* subset of the 20Newsgroup dataset

When we enhanced TLC using TLC++ by incorporating the labeled words in Table 11, Macro-F1 of text classification on the test dataset was increased to 0.840 from 0.680, which is 16% improvement in the performance over TLC. We also used the same labeled words in GE-FL algorithm, we observed that Macro-F1 was increased to 0.791 from 0.666. This is also evidence that performance of GE-FL can be improved when words are chosen based on topics. We want to emphasize here that, the human annotator only labeled a few topics and words and not a single document. When we compared performance of TLC++ algorithm with the semi-supervised text classification algorithm NB-EM [23], we observed that, in order to achieve similar performance on *pc-mac* dataset, it was necessary to label at least 350 documents. We used LingPipe[7] implementation of NB-EM algorithm in this experiment. However, in TLC++, significantly less labeling efforts were required. Hence, labeling a few words in addition to labeling topics can improve text classification performance.

5.5 Analysis of Quality of Topics

While assigning class labels to topics, we expect that the most probable words of a topic are coherent to a single theme (or class) and these words discriminate the topic from other topics. Mimno et al. [21] and Newman et al. [22] propose measures to evaluate coherence of topics. However, these measures do not evaluate how topics are different from each other.

Table 12 shows most probable words of topics inferred on the *med-space* subset of the 20Newsgroup dataset. We can observe here that the most probable words of each topic represent either *med (medical)* or *space*. On the other hand, topics inferred on the *pc-mac* dataset (Table 1) are not discriminative. In Table 9 we can observe that our algorithm performs better in *med-space* text classification task compared to *pc-mac*. Hence, we suspect that discriminative topics will perform better in text classification. We also suspect that when the classes are not clearly separable we get less discriminating topics.

We address these questions by measuring correlation between discrimination between topics and text classification performance.

To measure discrimination between topics we propose an information theoretic measure based on specific conditional entropy over a topic given a word and the remaining topics given the word. Our goal is to measure how a topic is different from other topics. Let t_{top} be a set of P most probable words of topic t. For each word $w \in t_{top}$ we compute specific conditional entropy of topic t given word w:

$$H(t|w) = -P_{t,w}log(P_{t,w}) - (1 - P_{t,w})log(1 - P_{t,w}) \quad (10)$$

ID	Most prominent words in the topic	Class (med / space)
0	msg **food doctor** day **pain** read **disease** problem **treatment** evidence	**med**
1	science **health** research information **medical** scientific water **cancer hiv aids**	**med**
2	**space nasa earth orbit** data **mission spacecraft lunar solar shuttle** system	**space**
3	**space launch nasa** cost technology **moon flight** big **satellite** vehicle billion	**space**

Table 12: Topic labeling on the *med-space* subset of the 20Newsgroup dataset

where,

$$P_{t,w} = \frac{\psi_{w,t}}{\sum_{i=1}^{T} \psi_{w,i}} \quad (11)$$

Here we assume that if a word is discriminating a topic from other topics then the word will have high probability of observing under the topic than that of rest of the topics and hence it will have smaller value of $H(t|w)$ than that of non-discriminating words.

We compute how a topic is different from other topics by taking average of $H(t|w)$ over top P most probable words of each topic (we empirically set P to 10).

$$H(t) = \frac{\sum_{w \in t_{top}} H(t|w)}{P} \quad (12)$$

We then use Equation 12 to compute overall discrimination between all topics of a topic model M_D ($H(M_D)$) by taking average of $H(t)$ over all topics weighted by probability of topic ($p(t)$) :

$$H(M_D) = \sum_{t=1}^{T} H(t)p(t) \quad (13)$$

We hypothesize that higher values of $H(M_D)$ indicate that topics are not discriminative and may lead to poor text classification performance. To verify this hypothesis, we compute $H(M_D)$ on different datasets discussed earlier and examine its correlation with performance of classifiers learned using TLC and SVM algorithms.

While learning a classifier using SVM, similar to Blei et al. [4], we learn a supervised SVM classifier using topics as features. Table 13 shows $H(M_D)$ and Macro-F1 of TLC and supervised SVM on different datasets discussed above. For the Reuters dataset, we consider average Macro-F1 of its 10 classification tasks and for SRAA dataset, we only consider "realauto-realaviation-simauto-simaviation" classification task, as "real-sim" and "auto-aviation" classification task use the same set of labeled topics.

We can observe here that Spearman's rank correlation coefficient between $H(M_D)$ and TLC is -0.758 and Spearman's rank correlation coefficient between $H(M_D)$ and SVM is -0.782. Both of these correlations are statistically significant using a two tailed paired t-test (p=0.05).

One reason behind the significantly negative correlation might be that when the classes are not clearly separable, most of the words in the vocabulary (other than stop-words) are shared by all the classes and these shared words have

[7] http://alias-i.com/lingpipe/demos/tutorial/em/read-me.html

Dataset	$H(M_D)$	TLC (Macro-F1)	SVM (Macro-F1)
pc-mac	0.402	0.680	0.673
med-space	0.367	0.943	0.935
autos-motorcycles-baseball-hockey	0.602	0.854	0.891
politics-religion	0.335	0.922	0.902
politics-sci	0.372	0.911	0.906
comp-religion-sci	0.383	0.871	0.875
politics-rec-religion-sci	0.439	0.858	0.863
realauto-realaviation-simauto-simaviation	0.648	0.806	0.824
Reuters	0.691	0.828	0.637
WebKB	0.621	0.684	0.738

Table 13: Experimental results ($H(M_D)$ and Macro-F1) of text classification and discrimination between topics of various datasets.

high frequency of occurrence, also they co-occur with discriminative words of all classes. As LDA uses higher order word co-occurrence while inferring topics [16], we get non-discriminating topics dominated by shared words.

If we compare *pc-mac* dataset and *med-space* dataset, then we can observe in Table 13 that $H(M_D)$ is higher for *pc-mac* dataset than that of *med-space* dataset and it also can be verified by observing topics inferred on both the datasets (Table 1 and 12). As per the quality of topics, we can say that *pc-mac* dataset is "harder" to classify than *med-space* dataset which is also evident from text classification performance using both TLC and SVM. Hence we accept our hypothesis that when the classes are not clearly separable we get less discriminating topics and hence poor text classification performance.

5.6 Discussion

Seifert et al. [26] propose an approach based on word clouds to generate training documents for text classification. They ask annotators to annotate condensed representation of documents, instead of the full text document for labeling single documents. These condensed representations are key sentences and key phrases, which are generated using graph based unsupervised algorithm- TextRank [20]. The key phrases are then represented as a tag cloud. Towards the goal of decreasing labeling time and cost for the generation of training documents, their evaluation shows that labeling key phrases is as fast and accurate as labeling full-text documents.

In Seifert et al. [26], annotators have to annotate key phrases in all documents in the corpus. However, in our algorithm annotators label only a few topics based on their top 30 most probable words or phrases, without reading a document in the corpus. Hence, our algorithm requires even less labeling efforts as compared to Seifert et al. [26].

Our work is motivated by two basic premises: Firstly, topic labelling is more effective than word labelling since topics are derived from corpus, and hence reflect statistical properties of words which are not accessible to a humans when she looks at words in seclusion. Also, topic labelling is easy since there are only a few topics. The second premise could be defeated in two ways : (a) The domain is hard, and hence the topics cannot be easily labelled based on their words (b) System- imposed constraints such as single label per topic (as in ClassifyLDA) could be prohibitive. We overcome the second limitation by the use of asymmetric priors, and the first by seeking very few word labels in addition to topic labels. The latter approach led to significant improvements in complex domains like pc-mac. We also show that word labels obtained using Information Gain from corpora labeled using topic labels can feed as a knowledge source and lead to improvements over the basic framework of LDA enriched by topic labels.

In general, our work is inspired by the philosophy that topics describe a domain fairly well at a high level and eliciting labels on just a few topics is an efficient way of making use of human expertise in that it could yield performance gains comparable to those that can be achieved using labelling large number of documents or alternately labeling words. In complex domains where topics are not coherent, or the association between topics and classes is not straightforward, we need to take the fallback option of increasing the granularity of labelling by insisting that few words are manually labelled as well. The idea of progressively going from general to specific also has a distant relative in transformation based learning which finds many applications in Natural Language Processing. To quote a fluid description [14] of the basic idea, "Often, learned rules are initially very general but become more specific as they approach the ground truth represented by the gold standard. An effective analogy is provided by Samuel [25], who attributes it to Terry Harvey. A painter can decide, upon painting a barnyard scene, to first colour everything blue, since the sky will comprise the majority of the canvas area. After the paint dries, he paints the barnyard which comprises a smaller area, but without taking care to avoid the windows, roof and doors, which will be painted more precisely at a later stage. Similarly, the first few rules can be applied over a broad area, increasing recall but also making many mistakes. As more rules are applied, some mistakes are corrected and precision generally goes up". That this idea of progressively going from general to specific is useful in practice is attested also by the fact that GE-FL benefits richly when words identified using topics (as opposed to those that are derived from document labels in the oracle approach) are labelled by humans.

6. CONCLUSIONS AND FUTURE WORK

In this paper we propose a weakly supervised text classification algorithm based on the generative property of unsupervised LDA. In our algorithm supervision comes in the form of labeling of a few LDA topics. So our algorithm does not need labeled documents and hence reduces cognitive load of annotators. We empirically demonstrate that application of asymmetric prior over document topic distribution and Topic-in-Set mechanism to deal with the topics representing multiple classes makes our algorithm robust in classification tasks like *pc-mac* where the classes are not clearly separable. Our results show that labeling topics is more helpful than labeling words, by overcoming challenges like coming up with a small set of statistically predictive as well as meaningful words to be labeled and dealing with polysemous words. The idea of topic labeling in TLC is further augmented in a scheme where annotators label a few words that are identified using feature selection over documents labeled using TLC. The resulting classifier outperforms state-of-the-art approaches aimed at reducing label acquisition overheads in text classification.

In future we would like to systematically evaluate cognitive loads in live user studies where annotators of different competence on domain participate in labeling topics or words. Finally, our approach and results give promising future research direction towards enabling human annotators to encode their domain knowledge into learning efficiently, exploring techniques to asses the complexity of a classification task, designing interfaces that will help users to interact with a dataset, and extending this approach beyond classification task.

7. REFERENCES

[1] D. Andrzejewski and X. Zhu. Latent Dirichlet Allocation With Topic-In-Set Knowledge. In *NAACL HLT*, pages 43–48, 2009.

[2] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional Word Clusters Vs. Words For Text Categorization. *J. Mach. Learn. Res.*, 3:1183–1208, Mar. 2003.

[3] D. M. Blei and J. D. McAuliffe. Supervised Topic Models. In *NIPS*, 2007.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, March 2003.

[5] A. Blum and T. Mitchell. Combining Labeled And Unlabeled Data With Co-Training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[6] S. Chakraborti, U. C. Beresi, N. Wiratunga, S. Massie, R. Lothian, and D. Khemani. Visualizing and Evaluating Complexity of Textual Case Bases. In *ECCBR*, pages 104–119, 2008.

[7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[8] G. Druck, G. Mann, and A. McCallum. Learning From Labeled Features Using Generalized Expectation Criteria. In *SIGIR*, pages 595–602, 2008.

[9] Y. Grandvalet and Y. Bengio. Semi-Supervised Learning By Entropy Minimization. In *NIPS*, 2004.

[10] T. L. Griffiths and M. Steyvers. Finding Scientific Topics. *PNAS*, 101(suppl. 1):5228–5235, April 2004.

[11] T. L. Griffiths, J. B. Tenenbaum, and M. Steyvers. Topics In Semantic Representation. *Psychological Review*, 114:2007, 2007.

[12] S. Hingmire, S. Chougule, G. K. Palshikar, and S. Chakraborti. Document Classification By Topic Labeling. In *SIGIR*, pages 877–880, 2013.

[13] T. Joachims. Transductive Inference For Text Classification Using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, 1999.

[14] G. Kotzé. Transformation-based tree-to-tree alignment. *Computational Linguistics in the Netherlands Journal*, 2:71–96, 2012.

[15] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative Learning For Dimensionality Reduction And Classification. In *NIPS*, 2008.

[16] S. Lee, J. Baker, J. Song, and J. C. Wetherbe. An Empirical Comparison Of Four Text Mining Methods. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, 2010.

[17] B. Liu, X. Li, W. S. Lee, and P. S. Yu. Text Classification By Labeling Words. In *Proceedings of the 19th national conference on Artifical intelligence*, pages 425–430, 2004.

[18] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[19] A. McCallum, G. Mann, and G. Druck. Generalized Expectation Criteria. Technical report, Department of Computer Science, University of Massachusetts Amherst, 2007.

[20] R. Mihalcea and P. Tarau. TextRank: Bringing Order Into Text. In *EMNLP*, pages 404–411, 2004.

[21] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing Semantic Coherence In Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272, 2011.

[22] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, 2010.

[23] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text Classification From Labeled And Unlabeled Documents Using Em. *Machine Learning - Special issue on information retrieval*, 39(2-3), May-June 2000.

[24] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers. Statistical Topic Models For Multi-Label Document Classification. *Mach. Learn.*, 88(1-2):157–208, 2012.

[25] K. Samuel. Lazy transformation-based learning. In *FLAIRS Conference*, pages 235–239, 1998.

[26] C. Seifert, E. Ulbrich, and M. Granitzer. Word Clouds For Efficient Document Labeling. In T. Elomaa, J. Hollmén, and H. Mannila, editors, *Discovery Science*, volume 6926, pages 292–306. Springer Berlin Heidelberg, 2011.

[27] M. Steyvers and T. Griffiths. Probabilistic Topic Models. In *Latent Semantic Analysis: A Road to Meaning*. 2006.

[28] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking LDA: Why Priors Matter. In *NIPS*, pages 1973–1981, 2009.

[29] Y. Yang and J. O. Pedersen. A Comparative Study On Feature Selection In Text Categorization. In *ICML*, pages 412–420, 1997.

[30] J. Zhu, A. Ahmed, and E. P. Xing. Medlda: Maximum Margin Supervised Topic Models For Regression And Classification. In *ICML*, pages 1257–1264, 2009.

[31] X. Zhu and Z. Ghahramani. Learning From Labeled And Unlabeled Data With Label Propagation. Technical report, Carnegie Mellon University, 2002.