LCCT: A Semi-supervised Model for Sentiment Classification

Min Yang[1] **Wenting Tu**[1] **Ziyu Lu**[1] **Wenpeng Yin**[2] **Kam-Pui Chow**[1]
[1]Department of Computer Science, The University of Hong Kong, Hong Kong
{myang,wttu,zylu,chow}@cs.hku.hk
[2]Center for Information and Language Processing, University of Munich, Germany
wenpeng@cis.lmu.de

Abstract

Analyzing public opinions towards products, services and social events is an important but challenging task. An accurate sentiment analyzer should take both lexicon-level information and corpus-level information into account. It also needs to exploit the domain-specific knowledge and utilize the common knowledge shared across domains. In addition, we want the algorithm being able to deal with missing labels and learning from incomplete sentiment lexicons. This paper presents a LCCT (**L**exicon-based and **C**orpus-based, **C**o-**T**raining) model for semi-supervised sentiment classification. The proposed method combines the idea of lexicon-based learning and corpus-based learning in a unified co-training framework. It is capable of incorporating both domain-specific and domain-independent knowledge. Extensive experiments show that it achieves very competitive classification accuracy, even with a small portion of labeled data. Comparing to state-of-the-art sentiment classification methods, the LCCT approach exhibits significantly better performances on a variety of datasets in both English and Chinese.

1 Introduction

Due to the popularity of opinion-rich resources (e.g., online review sites, forums, blogs and the microblogging websites), people express their opinions all over the Internet. Motivated by the demand of gleaning insights from such valuable data, a flurry of research devotes to the task of extracting people's opinions from online reviews. Such opinions could be expressed on products, services or policies, etc (Pang and Lee, 2008). Existing sentiment analysis approaches can be divided into two categories based on the source of information they use: the lexicon-based approach (Turney, 2002; Dave et al., 2003) and the corpus-based approach (Pang et al., 2002; Blitzer et al., 2007; Wan, 2009). The lexicon-based approach counts positive and negative terms in a review based on the sentiment dictionary and classifies the document as positive if it contains more positive terms than negative ones. On the contrary, the corpus-based approach uses supervised learning algorithms to train a sentiment classifier.

Further study (Kennedy and Inkpen, 2006; Andreevskaia and Bergler, 2008; Qiu et al., 2009) shows that corpus-based and lexicon-based approaches have complementary performances. Specifically, the corpus-based approach has high precision but low recall on positive instances, while the lexicon-based approach has high recall but low precision on positive instances. In fact, corpus-based approaches are over conservative in classifying instances as positive, because positive reviews usually contain many neutral statements. In contrast, the lexicon-based approaches tend to classify negative or neutral instances as positive when there are a few positive words appear in the document. It motivates us to develop a new approach that achieves good performance on both precision and recall evaluations.

Besides reviews on products and services, another rich source of opinion data are social reviews in forums, blogs and microblogging websites. Different from product reviews, the social reviews are not associated with numerical ratings, making it difficult to perform supervised classification. Since manual labeling is time consuming and expensive, it is

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 546–555,
Denver, Colorado, May 31 – June 5, 2015. ©2015 Association for Computational Linguistics

preferable to label a small portion of social reviews to perform semi-supervised learning, leveraging information from both labeled and unlabeled data.

In this paper, we propose a novel approach to handle the above two challenges. We presents the LCCT Model (**L**exicon-based and **C**orpus-based, **C**o-**T**raining Model), which treats the lexicon-based information and the corpus-based information as two views, and combine them via co-training (Blum and Mitchell, 1998). The algorithm naturally incorporates the framework of semi-supervised learning, as missing labels in each view can be estimated by the classifier trained from the other view. The proposed LCCT model exploits the complementary performance associated with the lexicon-based and the corpus-based approaches, taking the best of each side to improve the overall performance. We present a novel semi-supervised sentiment-aware LDA approach to build the lexicon-based classifier, which uses a minimal set of seed words (e.g., "good","happy" as positive seeds) as well as document sentiment labels to construct a domain-specific sentiment lexicon. This model reflects the domain-specific knowledge. We employ the stacked denoising auto-encoder (Vincent et al., 2008; Glorot et al., 2011) to build the corpus-based classifier. As Glorot et al. (Glorot et al., 2011) point out, the intermediate abstractions extracted in this way tend to reflect the domain-independent knowledge, unifying information across all domains. Finally, we use a co-training algorithm to combine the corpus-based and lexicon-based classifiers and to combine the domain-specific knowledge and the domain-independent knowledge.

The main contributions of our approach are three-folded. First, we propose a method that exploits both general domain-independent knowledge and specific domain-dependent knowledge, behaving like a human being when she analyzes the text. Second, we complement the lexicon-based approach and the corpus-based approach to overcome their respective classification biases. Third, our approach is capable of levering labeled and unlabeled data, unifying them into a semi-supervised learning framework. We conduct extensive experiments to verify the effectiveness of the proposed approach on real-world social reviews. The experiment results show that our model substantially outperforms the state-of-the-art methods in analyzing sentiments in online reviews.

2 Related Works

Sentiment analysis of natural language texts is an active research field. The papers by Pang and Lee (Pang and Lee, 2008) and Liu (Liu, 2012) describe most of the existing techniques for sentiment analysis and opinion mining. Sentiment analysis approaches can be categorized into lexicon-based approaches (Turney, 2002; Kennedy and Inkpen, 2006; Andreevskaia and Bergler, 2008) and corpus-based approaches (Pang et al., 2002; Blitzer et al., 2007; Wan, 2009). The lexicon-based approach uses a dictionary of opinion words (e.g., "good" and "bad") to identify the sentiment of a text. In contrast, the corpus-based approach can be seen as a statistical learning approach (Pang et al., 2002; Whitelaw et al., 2005; Wiebe and Riloff, 2005; Ye et al., 2009). The performance of corpus-based methods often degenerates when the labeled training data is insufficient.

As we have discussed earlier, corpus-based algorithms are overly conservative on positive reviews, while lexicon-based approaches are overly aggressive on positive reviews. There are several literature integrating both methods (Kennedy and Inkpen, 2006; Andreevskaia and Bergler, 2008; Qiu et al., 2009; Zhang et al., 2011). These methods require either a complete lexicon or a fully labeled corpus being available, which might not be true in practice. The method in this paper, in contrast, uses incomplete lexicon and partially labeled corpus as training examples.

On the other hand, there are semi-supervised methods in sentiment analysis which handle incomplete data (Wan, 2009; Dasgupta and Ng, 2009; Li et al., 2010; Zhou et al., 2010; Biyani et al., 2013). Nevertheless, none of them combines the lexicon-based and corpus-based approaches and thus they do not solve the bias problem in sentiment classification.

3 LCCT Model

In the LCCT model, we use a novel semi-supervised sentiment-aware LDA model to build the lexicon-based model. We use stacked denoising auto-encoder (Vincent et al., 2008; Glorot et al., 2011) to build the corpus-based model. Finally, a co-training algorithm is employed for semi-supervised

sentiment classification, and the two classifiers from corpus-based method and lexicon-based method are combined. The overall structure of the model is illustrated by Figure 1.

3.1 Lexicon-based Approach

For building the lexicon-based model, the key challenge is that a single word can carry multiple sentiment meanings in different domains, so that a general-purpose sentiment lexicon is less accurate than domain-specific lexicons. To solve this problem, we build a domain-specific sentiment lexicon by semi-supervised sentiment-aware LDA (ssLDA). The ssLDA method takes semi-supervised data as input.

3.1.1 Semi-supervised Sentiment-aware LDA

In this section, we describe how each word of the corpus is generated by the ssLDA model, then illustrate its inference method. Each document has three classes of topics: $K^{(p)}$ positive sentiment topics, $K^{(n)}$ negative sentiment topics, and $K^{(u)}$ neutral sentiment topics. Each document is a mixture of the three classes of topics. Each topic is associated with a multinomial distribution over words. To prevent conceptual confusion, we use a superscript "(p)" and "(n)" to indicate variables relating to positive and negative sentiment topics, and a superscript "(u)" to indicate variables relating to neutral sentiment topics. In addition, we assume that the vocabulary consists of V distinct words indexed by $\{1, \ldots, V\}$.

For each word w, there is a multinomial distribution determining which class of topics that w belongs to. This prior distribution is sampled from a Dirichlet distribution $\mathrm{Dir}(\lambda)$, where $\lambda = (\lambda^{(p)}, \lambda^{(n)}, \lambda^{(u)})$ is a vector of three scalars. For documents with different sentiment labels, we choose different values of λ, so that words in the document with a positive label has a higher probability belonging to positive topics, and vice versa. In the semi-supervised setting, a document usually doesn't have a sentiment label. In that case, the value of λ is equal to $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Given the class of topics, there is another multinomial distribution indicating the particular topic that the word belongs to. If it turns out that the word belongs to a positive sentiment class, then its topic distribution is drawn from a biased Dirichlet prior $\phi_w^{(p)} \sim \mathrm{Dir}(\beta_w^{(p)})$. The vector $\beta_w^{(p)} \in \mathbb{R}^V$ is constructed by

$$\beta_{w,k}^{(p)} := \gamma_0(1 - \omega_w) + \gamma_1\omega_w \quad \text{for } k \in \{1, \ldots, K\} \tag{1}$$

We set $\omega_w = 1$ if the word w is a positive seed word, otherwise, we set $\omega_w = 0$. The scalars γ_0 and γ_1 are hyperparameters. Intuitively, the biased prior enforces a positive seed word more probably drawn from a positive sentiment topic. The distributions $\phi_{\mathrm{w}}^{(\mathrm{n})} \sim \mathrm{Dir}(\beta_w^{(n)})$ and $\phi_w^{(u)} \sim \mathrm{Dir}(\beta_w^{(u)})$ for negative and neutral sentiment topics are similarly constructed. Once the topic is determined, the word is generated from a multinomial distribution that associates with the topic. We summarize the generative process of the ssLDA model as below:

1. For each word w in the vocabulary, draw the distributions of topics for three sentiment classes: $\phi_w^{(p)} \sim \mathrm{Dir}(\beta_w^{(p)})$, $\phi_w^{(n)} \sim \mathrm{Dir}(\beta_w^{(n)})$ and $\phi_w^{(u)} \sim \mathrm{Dir}(\beta_w^{(u)})$.

2. For each topic k, draw the distribution over words: $\theta_k^{(p)} \sim \mathrm{Dir}(\alpha)$, $\theta_k^{(n)} \sim \mathrm{Dir}(\alpha)$ and $\theta_k^{(u)} \sim \mathrm{Dir}(\alpha)$.

3. For each document in the corpus
   (a) Draw sentiment class distribution p from either $\mathrm{Dir}(\lambda^{(p)})$, $\mathrm{Dir}(\lambda^{(n)})$ or $\mathrm{Dir}(\lambda^{(u)})$ based on the document's sentiment label.
   (b) For each word in document, Draw sentiment class indicator $c \sim \mathrm{Mult}(p)$, then generate the word's topic z from $\mathrm{Mult}(\phi_w^{(c)})$, and generate the word w from $\mathrm{Mult}(\theta_z^{(c)})$.

Given hyper-parameters α, λ, and $\{\beta^{(s)}, \beta^{(n)}, \beta^{(u)}\}$, our goal is to estimate the latent variables in the ssLDA model. We present a collapsed Gibbs-sampling algorithm, which iteratively takes a word w from the corpus and samples the topic that the word belongs to. The reader may refer to (Yang et al., 2014) for a detailed derivation of the sampling procedure. Let the whole corpus excluding the current word be denoted by D. Let $n_{i,w}^{(p)}$ (or $n_{j,w}^{(n)}$, or $n_{k,w}^{(u)}$) indicate the number of occurrences of positive sentiment topic $i^{(p)}$ (or negative sentiment topic $j^{(n)}$, or neutral sentiment topic $k^{(u)}$) with word w in

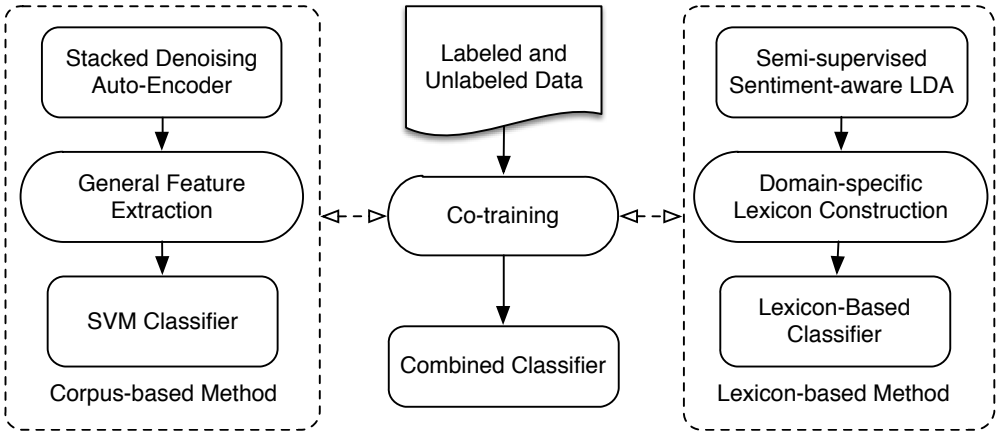

Figure 1: Algorithm Overview

the whole corpus. Let $m_i^{(p)}$ (or $m_j^{(n)}$, or $m_k^{(u)}$) indicate the number of occurrence of positive sentiment topic $i^{(p)}$ (or negative sentiment topic $j^{(n)}$, or neutral sentiment topic $k^{(u)}$) in the current document. Then, the posterior probability that the current word w belongs to a specific topic is presented as follow

$$\Pr\left(z = i^{(p)}|D\right) \propto (\lambda^{(p)} + \sum_{i=1}^{K^{(p)}} m_i^{(p)}) \cdot \frac{\alpha + m_i^{(p)}}{K^{(p)}\alpha + \sum_{i'=1}^{K^{(p)}} m_{i'}^{(p)}} \cdot \frac{\beta_{i,w}^{(p)} + n_{i,w}^{(p)}}{\sum_{w'=1}^{V}\left(\beta_{i,w'}^{(p)} + n_{i,w'}^{(p)}\right)} \quad (2)$$

$$\Pr\left(z = j^{(n)}|D\right) \propto (\lambda^{(p)} + \sum_{i=1}^{K^{(p)}} m_i^{(p)}) \cdot \frac{\alpha + m_j^{(n)}}{K^{(n)}\alpha + \sum_{j'=1}^{K^{(n)}} m_{j'}^{(n)}} \cdot \frac{\beta_{j,w}^{(n)} + n_{i,w}^{(n)}}{\sum_{w'=1}^{V}\left(\beta_{j,w'}^{(n)} + n_{j,w'}^{(n)}\right)} \quad (3)$$

$$\Pr\left(z = k^{(u)}|D\right) \propto (\lambda^{(p)} + \sum_{i=1}^{K^{(p)}} m_i^{(p)}) \cdot \frac{\alpha + m_k^{(u)}}{K^{(u)}\alpha + \sum_{k'=1}^{K^{(u)}} m_{u'}^{(n)}} \cdot \frac{\beta_{k,w}^{(u)} + n_{k,w}^{(u)}}{\sum_{w'=1}^{V}\left(\beta_{k,w'}^{(u)} + n_{k,w'}^{(u)}\right)} \quad (4)$$

By equations (2), (3), and (4), we can sample the topic z for each word. In the Gibbs sampling procedure, we only need to maintain the counters $n^{(p)}$, $n^{(n)}$, $n^{(u)}$, $m^{(p)}$, $m^{(n)}$ and $m^{(u)}$, which takes $O(1)$ time to update for each iteration.

3.1.2 Lexicon Construction and Sentiment Classification

Once we obtain the topic of each word, we obtain the value of hidden variables $p^{(c)}, \theta^{(c)}, \phi^{(c)}$, where $c \in \{p, n, u\}$. The goal is to use these values to construct a sentiment lexicon, which assigns sentiment scores to each word. In particular, we need the probability that each word w appears in a certain sentiment class, i.e. we want to calculate $\Pr(c \in \{p, n, u\}|w)$ for the sentiment indicator c. We use $\gamma_w^{(p)}$, $\gamma_w^{(n)}$, $\gamma_w^{(u)}$ to represent these probabilities. By the ssLDA's model specification, we define

$$\gamma_w^{(p)} := \Pr(c = p|w) \propto p^{(p)} \cdot \sum_{i=1}^{K^{(p)}} \theta_{i,w}^{(p)} \phi_{w,i}^{(p)} \quad (5)$$

$$\gamma_w^{(n)} := \Pr(c = n|w) \propto p^{(n)} \cdot \sum_{j=1}^{K^{(n)}} \theta_{i,w}^{(n)} \phi_{w,j}^{(n)} \quad (6)$$

$$\gamma_w^{(u)} := \Pr(c = u|w) \propto p^{(u)} \cdot \sum_{k=1}^{K^{(u)}} \theta_{i,w}^{(u)} \phi_{w,k}^{(u)} \quad (7)$$

We construct the sentiment lexicon for each word w by comparing $\gamma_w^{(p)}$, $\gamma_w^{(n)}$ and $\gamma_w^{(u)}$. If $\gamma_w^{(p)}$ is the greatest value, then the word w is considered to convey positive sentiment, and is added to the positive sentiment lexicon with weight $\gamma_w^{(p)}$. If $\phi_{1,w}^{(s)}$ is the greatest, then the word w is added to the negative sentiment lexicon with weight $-\gamma_w^{(n)}$. Otherwise, the word w is considered neutral and not included in the sentiment lexicon.

It remains to classify the sentiment for each document. We aggregate the weights for each word, so that the document is classified as "positive" if the accumulated weight is larger than zero; Otherwise,

it is classified as "negative". The proposed model is a semi-supervised method since it is capable of processing documents without the sentiment label. This property makes the proposed method suitable for co-training.

3.2 Corpus-based Method

The deep learning approach, especially Stacked Denoising Auto-encoders (SDA), has been shown highly beneficial for extracting domain-independent knowledge (Glorot et al., 2011). Thus, we use SDA to construct the corpus-based sentiment classifier. The stacked autoencoder method was introduced by Rumelhart, Hinton and Williams (Rumelhart et al., 1985) and its denoising variant was proposed by Vincent et al. (Vincent et al., 2010). Recently, it has become an essential building block in deep learning architectures. A basic denoising autoencoder consists of an input layer, a hidden layer and an output layer. The procedure can be interpreted into two phases, i.e., encode and decode. In the encoding phrase, an encoder function is employed to map input data into a feature vector h. For each sample x from input dataset $\{x^{(1)}, \ldots, x^{(N)}\}$, we have

$$h = f(U^T(x + \epsilon) + b) \tag{8}$$

where $f(x)$ is sigmoid activation function, U is the weight matrix between input layer and hidden layer, b_h is the bias of each input layer neuron and ϵ is a random Gaussian noise. In the decoding phrase, a decoder function is deployed to remap the feature vector in the feature space back to the input space, producing a reconstruction $\hat{x}$. The decoder function takes the following form

$$\hat{x} = f(V^T h + b') \tag{9}$$

where $f(x)$ is also a sigmoid function, V is the weight matrix between the hidden layer and the output layer, and b' is the bias. The parameters of the SDA models, namely $\theta = \{U, V, b, b'\}$, are learned by minimizing the reconstruction error $L(x, \hat{x})$ over all training instances:

$$J(\theta) = \sum_{x^{(t)}} L(x^{(t)}, \hat{x}^{(t)}) \tag{10}$$

where $L(\cdot, \cdot)$ is measure of discrepancy. Popular choices of L include squared error and Kullback-Liebler divergence. By iteratively adding autoencoders on top of a trained denoising autoencoder, we obtain the stacked denoising autoencoder (SDA). Once trained, their parameters can be used to initialize a supervised learning algorithm. In this paper, SDA is learnt in a greedy layer-wise fashion using stochastic gradient descent. For the first layer, the decoder is activated by a sigmoid function, and the Kullback-Liebler divergence is used as the reconstruction error. For the remaining layers, we use the softplus function for activation. After the SDA parameters are trained (on both labeled and unlabeled data) and the high-level representation of each data instance is obtained, a SVM classifier is employed using the resulting representation (of labeled data) to train a sentiment classifier.

3.3 Combining two Methods with Co-training

Algorithm 1 Co-training with corpus-based and lexicon-based methods

- Inputs: labeled training data $\boldsymbol{L}$, unlabeled training data $\boldsymbol{U}$
- Create a pool $\boldsymbol{U}'$ of examples by choosing u unlabeled examples at random, then loop for k iterations
  - use $\boldsymbol{L}$ and $\boldsymbol{U}$ to train a corpus-based classifier f_1, then use f_1 to label samples from $\boldsymbol{U}'$. Let $\boldsymbol{A}_1$ be the set of p positive and n negative most confidently labeled examples.
  - use $\boldsymbol{L}$ and $\boldsymbol{U}$ to train a lexicon-based classifier f_2, then use f_2 to label samples from $\boldsymbol{U}'$. Let $\boldsymbol{A}_2$ be the set of p positive and n negative most confidently labeled examples.
  - Add f_1 and f_2 to the set $\boldsymbol{C}$ of classifiers and add the self-labeled examples $\boldsymbol{A}_1 \cup \boldsymbol{A}_2$ to the labeled dataset $\boldsymbol{L}$. Randomly choose $2p + 2n$ examples from $\boldsymbol{U}$ to replenish $\boldsymbol{U}'$
- For testing, run all classifiers in $\boldsymbol{C}$ and output the majority vote.

We employ a variant of co-training algorithm to train the classifier with a small number of labeled data and a large number of unlabeled data. The co-training approach is well known for semi-supervised approach (Blum and Mitchell, 1998). For our problem, the two views of co-training are lexicon-based method (domain-specific knowledge) and corpus-based method (domain-independent knowledge). Initially, both classifiers are trained with the partially available labels, as described by the above two subsections. Then, we use one of the two classifiers to label the unlabeled documents, adding its labels to

the pool of labeled data, re-training the other classifier using the new labeled data. The procedure is performed iteratively. After a sufficient number of iterations, we obtain a set of classifiers and we combine them using a majority-voting scheme to predict the sentiment label for test data. The details of the algorithm are summarized in Algorithm 1.

4 Experiments

In this section, we compare the proposed LCCT model with state-of-the-art methods in sentiment classification. The experiment demonstrates the superior performance of our approach.

4.1 Datasets

We conduct experiments on English and Chinese reviews from three datasets. In this subsection, we describe the datasets.

Movie Review (MR) dataset in English The movie reviews are selected if the rating was stars or a numerical score. In this paper, we use the Movie Review dataset containing 1000 positive examples and 1000 negative examples (Pang and Lee, 2004). Positive labels were assigned to reviews that had a rating above 3.5 stars and negative labels were assigned to the rest (Pang and Lee, 2004).

SemEval-2013 (SemEval) dataset in English This dataset is constructed for the Twitter sentiment analysis task (Task 2) in the Semantic Evaluation of Systems challenge (SemEval-2013). All the tweets were manually annotated by 5 Amazon Mechanical Turk workers with negative, positive and neutral labels. SemEval contains 13,975 tweets with 2,186 negative, 6,440 neutrals and 5,349 positives tweets. We collect the 2,186 negative tweets and 5,349 positive tweets as the training data.

COAE-2009 (COAE) dataset in Chinese This dataset is provided by COAE 2009 [1] (Task 4). The corpus consists of 39,976 documents and 50 topics. The topics cover education, entertainment, finance, computer, etc. In this paper, we select the 2202 negative and 1248 positive documents as our dataset.

In all experiments, data preprocessing is performed. For English dataset, the texts are first tokenized using the natural language toolkit NLTK[2].

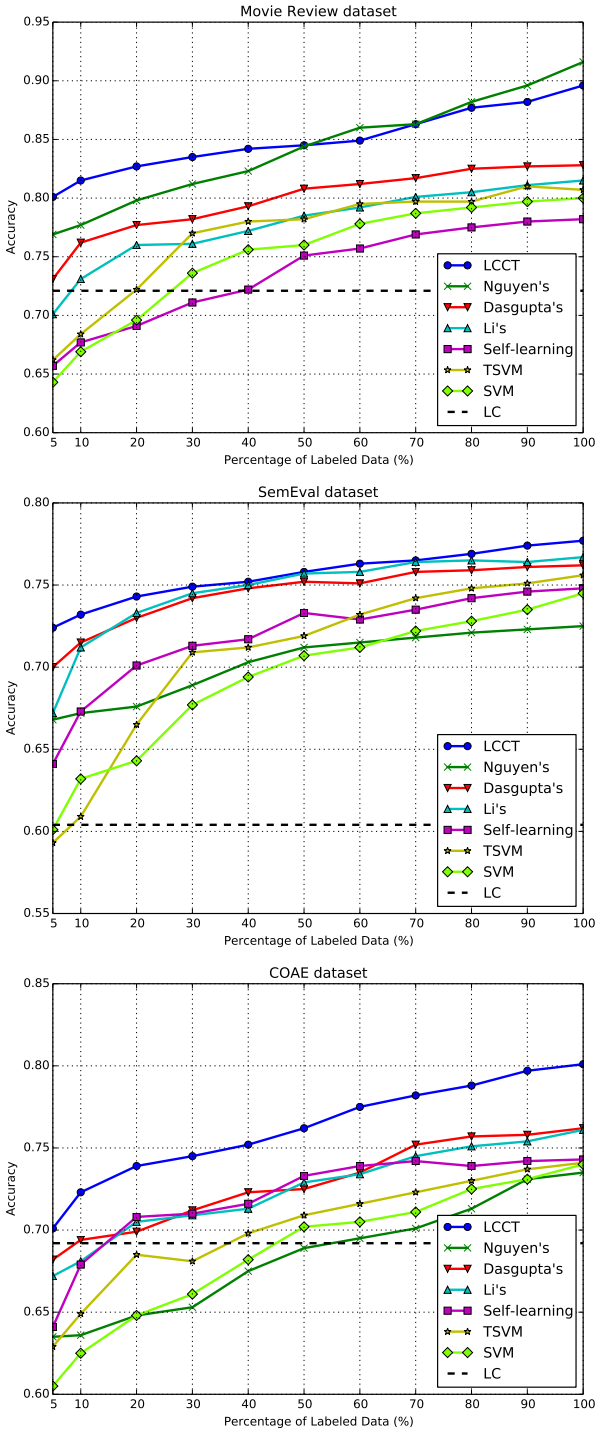

Figure 2: Comparing classification accuracy by varying the percentage of labeled data from 5% to 100%. The LCCT model is robust to incomplete data.

[1] http://ir-china.org.cn/coae2009.html
[2] http://www.nltk.org

Then, we remove non-alphabet characters, numbers, pronoun, punctuation and stop words from the text. Finally, the WordNet stemmer[3] is applied to reduce the vocabulary size and settle the issue of data sparseness. For Chinese dataset, we first perform Chinese word segmentation with a popular Chinese auto-segmentation system ICTCLAS[4]. Then, the words about time, numeral words, pronoun and punctuation are removed as they are unrelated to the sentiment analysis task.

4.2 Implementation Details

We specify the hyper-parameters we use for the experiments. For all datasets, we choose $\alpha = 0.5$, $\lambda^{(p)} = (0.95, 0.25, 0.4)$, $\lambda^{(n)} = (0.25, 0.95, 0.4)$, $\lambda^{(u)} = (0.6, 0.6, 0.4)$ and $(\gamma_0, \gamma_1) = (0.25, 0.75)$. We use cross-validation to set the number of topics on datasets MR, SemEval and COAE as 20, 10 and 20, respectively. The seed words used to construct English and Chinese lexicons are the same as in previous literatures (Xie and Li, 2012) and (Yang et al., 2014). For the corpus-based method, each document is transformed into binary vectors which encodes the presence/absence of the terms. The autoencoder is constructed with 500 input neurons and 200 hidden neurons. Each autoencoder is trained by back propagation with 400 iterations.

For all datasets, we set the iteration number of co-training to be $k = 50$. Other parameters of co-training are chosen by cross-validation: u is set to be 10% of all unlabeled data, the sum of p and n are 0.8% of all unlabeled data, while their ratio are determined by the ratio of positive and negative samples in labeled training data.

4.3 Baseline Methods

In this paper, we evaluate and compare our approach with an unsupervised method, two supervised methods and a variety of semi-supervised methods:

SVM: 5000 words with greatest information gain are chosen as features. In our experiment, we use the LibLinear[5] implementation of SVM.

Lexical Classifier (LC): This method calculates the number of positive words and negative words contained in the Opinion Lexicon (Hu and Liu, 2004) for English texts or the HowNet[6] lexicon for Chinese texts. If the positive sentiment words are more than negative words, then the document is classified as positive, and vice versa.

Self-learning: Following the idea of (Zhu, 2006), this method uses the unlabeled data in a bootstrapping way. The SVM classifier is used to select most confident unlabeled samples in each iteration.

Transductive SVM (TSVM) : Following the idea of (Joachims, 1999), this method seeks the largest separation between labeled and unlabeled data through regularization. We implement it with the SVM-light toolkit [7].

Dasgupta's method: This is a popular semi-supervised approach to automatic sentiment classification proposed by Dasgupta and Ng (Dasgupta and Ng, 2009). The unambiguous reviews are first mined using spectral techniques, then classified by a combination of active learning, transductive learning, and ensemble learning.

Li's method: This method is proposed in (Li et al., 2010). An unsupervised bootstrapping method is adopted to automatically split documents into personal and impersonal views. Then, two views are combined by an ensemble of individual classifier generated by each view. The co-training algorithm is utilized to incorporate unlabeled data.

Nguyen's method: This method is proposed in (Nguyen et al., 2014), which achieves the state-of-the-art results in supervised sentiment classification. We follow all the settings in (Nguyen et al., 2014). For the document with no associated score, we predict a score for the document as the values of the rating-based features using a regression model learned from SRA14[8] dataset.

4.4 Experiment Results

For each dataset, we use 80% instances as the training data and the remaining are used for testing. To test the performance of semi-supervised learning, we randomly select 10% of the training instances as labeled data and treat the remaining as unlabeled. For fair comparison, the fully supervised SVM and Nguyen's method use the 10% labeled data for training.

[3]http://wordnet.princeton.edu/

[4]http://www.ictclas.org

[5]http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[6]http://www.keenage.com/download/sentiment.rar

[7]http://svmlight.joachims.org/

[8]https://sites.google.com/site/nquocdai/resources

Dataset	SVM	LC	Self-learning	TSVM	Dasgupta's	Li's	Nguyen's	LCCT
MR	0.669	0.721	0.677	0.684	0.762	0.731	0.769	0.815
SemEval	0.632	0.604	0.675	0.609	0.735	0.702	0.652	0.775
COAE	0.625	0.706	0.679	0.649	0.709	0.692	0.642	0.713

Table 1: Comparing classification accuracy with 10% labeled data. The LCCT model performs significantly better

We summarize the experiment results in Table 1. According to Table 1, the proposed LCCT method substantially and consistently outperforms other methods on all the three datasets. This verifies the effectiveness of the proposed approach and demonstrates its advantage in semi-supervised sentiment analysis where reviews are from different domains and different language. For example, the overall accuracy of our algorithm is 5.3% higher than Dasgupta's method and 13.1% higher than TSVM on Movie Reviews dataset. On other datasets, we observe the similar results. To verify that unlabeled data improves the performance, we compare the SVM and Nguyen's classifier trained on 10% of the labeled data with other semi-supervised classifiers. Table 1 shows that the semi-supervised learning methods greatly benefit from using unlabeled data, especially on the Movie Reviews and on the SemEval dataset. Surprisingly, on the COAE dataset, lexicon-based method turns out to outperform SVM, self-learning and TSVM. The reason might be that the topics in the COAE dataset are pretty diverse. Without sufficient labeled data or prior knowledge such as sentiment lexicon, the corpus-based classifiers tend to separate the documents into topical sub-clusters as opposed to sentiment classes.

To understand the performance of our algorithm with respect to different portions of labeled data, we compare our algorithm with baseline methods by varying the percentage of labeled data from 5% to 100%. Figure 2 shows that our approach is robust and achieves excellent performance on different labeling percentages. As expected, having more labeled data improves the performance. The LCCT method achieves a relative high accuracy with 10% of the reviews labeled, better than SVM, TSVM and Self-learning with 100% of the reviews labeled. On the other hand, when all the training data are labeled, LCCT is still significantly more accurate than all the competitors except Nguyen's method. Although, the accuracy of Nguyen's method is slightly better than ours on Movie Reviews dataset, it dosen't perform well on SemEval and COAE datasets since the rating-based features learned from score-associated product reviews cannot significantly benefit the social reviews in forums and blogs, etc. The main advantage of our model comes from its capability of exploiting the complementary information from the lexicon-based approach and the corpus-based approach. Another reason for the effectiveness of our approach is the way that we combine the domain-independent knowledge and the domain-specific knowledge.

It is known that both the corpus-based approach and the lexicon-based approach have classification biases (Kennedy and Inkpen, 2006; Andreevskaia and Bergler, 2008; Qiu et al., 2009). To evaluate the effectiveness of our algorithm in reducing the bias, we compare it with the classifier that only uses one view of the LCCT model: either using the corpus-based view or using the lexicon-based view. The comparison is conducted on the Movie Review dataset. As Table 2 shows, our algorithm achieves good performance on both precision and recall. In contrast, the baseline methods either have high precision but low recall, or have high recall but low precision. The experiment result suggests that combining the two views is essential in eliminating the classification bias.

Data	Corpus-based		Lexicon-based		LCCT	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
MR pos.	0.92	0.79	0.67	0.86	0.90	0.86
MR neg.	0.78	0.90	0.80	0.58	0.88	0.89

Table 2: Precision and recall on Movie reviews

5 Conclusions

We have proposed the LCCT model for semi-supervised sentiment classification, combining the idea of lexicon-based learning and corpus-based learning in a unified co-training framework. It is capable of incorporating both domain-specific and domain-independent knowledge. Comparing to state-of-the-art sentiment classification methods, the LCCT approach exhibits significantly better performances on a variety of datasets in both English and Chinese, even with a small portion of labeled data.

References

Alina Andreevskaia and Sabine Bergler. 2008. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *ACL*, pages 290–298.

Prakhar Biyani, Cornelia Caragea, Prasenjit Mitra, Chong Zhou, John Yen, Greta E Greer, and Kenneth Portier. 2013. Co-training over domain-independent and domain-dependent features for sentiment analysis of an online cancer support community. In *the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 413–417. ACM.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100. ACM.

Sajib Dasgupta and Vincent Ng. 2009. Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *ACL-IJCNLP:Volume 2*, pages 701–709. Association for Computational Linguistics.

Kushal Dave, Steve Lawrence, and David M Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528. ACM.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *SIGKDD*, pages 168–177. ACM.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209.

Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.

Shoushan Li, Chu-Ren Huang, Guodong Zhou, and Sophia Yat Mei Lee. 2010. Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *ACL*, pages 414–423. Association for Computational Linguistics.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Dai Quoc Nguyen, Dat Quoc Nguyen, Thanh Vu, and Son Bao Pham. 2014. Sentiment classification on polarity reviews: an empirical study using rating-based features.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, page 271. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP: Volume 10*, pages 79–86. Association for Computational Linguistics.

Likun Qiu, Weishi Zhang, Changjian Hu, and Kai Zhao. 2009. Selc: a self-supervised model for sentiment classification. In *CIKM*, pages 929–936. ACM.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, DTIC Document.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424. Association for Computational Linguistics.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103. ACM.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 9999:3371–3408.

Xiaojun Wan. 2009. Co-training for cross-lingual sentiment classification. In *ACL-IJCNLP: Volume 1-Volume 1*, pages 235–243. Association for Computational Linguistics.

Casey Whitelaw, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *CIKM*, pages 625–631. ACM.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Computational Linguistics and Intelligent Text Processing*, pages 486–497. Springer.

Rui Xie and Chunping Li. 2012. Lexicon construction: A topic model approach. In *International Conference on Systems and Informatics (ICSAI)*, pages 2299–2303. IEEE.

Min Yang, Dingju Zhu, Rashed Mustafa, and Kam-Pui Chow. 2014. Learning domain-specific sentiment lexicon with supervised sentiment-aware lda. *ECAI 2014*, pages 927–932.

Qiang Ye, Ziqiong Zhang, and Rob Law. 2009. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, 36(3):6527–6535.

Ley Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. 2011. Combining lexiconbased and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89.

Shusen Zhou, Qingcai Chen, and Xiaolong Wang. 2010. Active deep networks for semi-supervised sentiment classification. In *Coling: Posters*, pages 1515–1523. Association for Computational Linguistics.

Xiaojin Zhu. 2006. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3.