Lifelong Learning for Sentiment Classification

Zhiyuan Chen, Nianzu Ma, Bing Liu
Department of Computer Science
University of Illinois at Chicago
{czyuanacm,jingyima005}@gmail.com,liub@cs.uic.edu

Abstract

This paper proposes a novel lifelong learning (LL) approach to sentiment classification. LL mimics the human continuous learning process, i.e., retaining the knowledge learned from past tasks and use it to help future learning. In this paper, we first discuss LL in general and then LL for sentiment classification in particular. The proposed LL approach adopts a Bayesian optimization framework based on stochastic gradient descent. Our experimental results show that the proposed method outperforms baseline methods significantly, which demonstrates that lifelong learning is a promising research direction.

1 Introduction

Sentiment classification is the task of classifying an opinion document as expressing a positive or negative sentiment. Liu (2012) and Pang and Lee (2008) provided good surveys of the existing research. In this paper, we tackle sentiment classification from a novel angle, *lifelong learning* (LL), or *lifelong machine learning*. This learning paradigm aims to learn as humans do: retaining the learned knowledge from the past and use the knowledge to help future learning (Thrun, 1998, Chen and Liu, 2014b, Silver et al., 2013).

Although many machine learning topics and techniques are related to LL, e.g., lifelong learning (Thrun, 1998, Chen and Liu, 2014b, Silver et al., 2013), transfer learning (Jiang, 2008, Pan and Yang, 2010), multi-task learning (Caruana, 1997), never-ending learning (Carlson et al., 2010), self-taught learning (Raina et al., 2007), and online learning (Bottou, 1998), there is still no unified definition for LL.

Based on the prior work and our research, to build an LL system, we believe that we need to answer the following key questions:

1. What information should be retained from the past learning tasks?
2. What forms of knowledge will be used to help future learning?
3. How does the system obtain the knowledge?
4. How does the system use the knowledge to help future learning?

Motivated by these questions, we present the following definition of *lifelong learning* (LL).

Definition (Lifelong Learning): A learner has performed learning on a sequence of tasks, from 1 to $N-1$. When faced with the Nth task, it uses the knowledge gained in the past $N-1$ tasks to help learning for the Nth task. An LL system thus needs the following four general components:

1. *Past Information Store* (*PIS*): It stores the information resulted from the past learning. This may involve sub-stores for information such as (1) the original data used in each past task, (2) intermediate results from the learning of each past task, and (3) the final model or patterns learned from the past task, respectively.
2. *Knowledge Base* (*KB*): It stores the knowledge mined or consolidated from PIS (Past Information Store). This requires a knowledge representation scheme suitable for the application.
3. *Knowledge Miner* (*KM*). It mines knowledge from PIS (Past Information Store). This mining can be regarded as a meta-learning process because it learns knowledge from information resulted from learning of the past tasks. The knowledge is stored to KB (Knowledge Base).
4. *Knowledge-Based Learner* (*KBL*): Given the knowledge in KB, this learner is able to leverage the knowledge and/or some information in PIS for the new task.

Based on this, we can define *lifelong sentiment classification* (LSC):

Definition (Lifelong Sentiment Classification): A learner has performed a sequence of supervised

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 750–756,
Beijing, China, July 26-31, 2015.

sentiment classification tasks, from 1 to $N-1$, where each task consists of a set of training documents with positive and negative polarity labels. Given the Nth task, it uses the knowledge gained in the past $N-1$ tasks to learn a better classifier for the Nth task.

It is useful to note that although many researchers have used transfer learning for supervised sentiment classification, LL is different from the classic transfer learning or domain adaptation (Pan and Yang, 2010). Transfer learning typically uses labeled training data from one (or more) source domain(s) to help learning in the target domain that has little or no labeled data (Aue and Gamon, 2005, Bollegala et al., 2011). It does not use the results of the past learning or knowledge mined from the results of the past learning. Further, transfer learning is usually inferior to traditional supervised learning when the target domain already has good training data. In contrast, our target (or future) domain/task has good training data and we aim to further improve the learning using both the target domain training data and the knowledge gained in past learning. To be consistent with prior research, we treat the classification of one domain as one learning task.

One question is why the past learning tasks can contribute to the target domain classification given that the target domain already has labeled training data. The key reason is that the training data may not be fully representative of the test data due to the *sample selection bias* (Heckman, 1979, Shimodaira, 2000, Zadrozny, 2004). In few real-life applications, the training data are fully representative of the test data. For example, in a sentiment classification application, the test data may contain some sentiment words that are absent in the training data of the target domain, while these sentiment words have appeared in some past domains. So the past domain knowledge can provide the prior polarity information in this situation.

Like most existing sentiment classification papers (Liu, 2012), this paper focuses on binary classification, i.e., positive ($+$) and negative ($-$) polarities. But the proposed method is also applicable to multi-class classification. To embed and use the knowledge in building the target domain classifier, we propose a novel optimization method based on the Naïve Bayesian (NB) framework and stochastic gradient descent. The knowledge is incorporated using penalty terms in the optimization formulation. This paper makes three contributions:

1. It proposes a novel lifelong learning approach to sentiment classification, called *lifelong sentiment classification* (LSC).
2. It proposes an optimization method that uses penalty terms to embed the knowledge gained in the past and to deal with domain dependent sentiment words to build a better classifier.
3. It creates a large corpus containing reviews from 20 diverse product domains for extensive evaluation. The experimental results demonstrate the superiority of the proposed method.

2 Related Work

Our work is mainly related to lifelong learning and multi-task learning (Thrun, 1998, Caruana, 1997, Chen and Liu, 2014b, Silver et al., 2013). Existing lifelong learning approaches focused on exploiting invariances (Thrun, 1998) and other types of knowledge (Chen and Liu, 2014b, Chen and Liu, 2014a, Ruvolo and Eaton, 2013) across multiple tasks. Multi-task learning optimizes the learning of multiple related tasks at the same time (Caruana, 1997, Chen et al., 2011, Saha et al., 2011, Zhang et al., 2008). However, these methods are not for sentiment analysis. Also, our naïve Bayesian optimization based LL method is quite different from all these existing techniques.

Our work is also related to transfer learning or domain adaptation (Pan and Yang, 2010). In the sentiment classification context, Aue and Gamon (2005) trained sentiment classifiers for the target domain using various mixes of labeled and unlabeled reviews. Blitzer et al. (2007) proposed to first find some common or pivot features from the source and the target, and then identify correlated features with the pivot features. The final classifier is built using the combined features. Li and Zong (2008) built a meta-classifier (called CLF) using the outputs of each base classifier constructed in each domain. Other works along similar lines include (Andreevskaia and Bergler, 2008, Bollegala et al., 2011, He et al., 2011, Ku et al., 2009, Li et al., 2012, Li et al., 2013, Pan and Yang, 2010, Tan et al., 2007, Wu et al., 2009, Xia and Zong, 2011, Yoshida et al., 2011). Additional details about these and other related works can be found in (Liu, 2012). However, as we discussed in the introduction, these methods do not focus on the ability to accumulate learned knowledge and leverage it in new learning in a lifelong manner.

3 Proposed LSC Technique

3.1 Naïve Bayesian Text Classification

Before presenting the proposed method, we briefly review the Naïve Bayesian (NB) text classification as our method uses it as the foundation.

NB text classification (McCallum and Nigam, 1998) basically computes the conditional probability of each word w given each class c_j (i.e., $P(w|c_j)$) and the prior probability of each class c_j (i.e., $P(c_j)$), which are used to calculate the posterior probability of each class c_j given a test document d (i.e., $P(c_j|d)$). c_j is either positive $(+)$ or negative $(-)$ in our case.

The key parameter $P(w|c_j)$ is computed as:

$$P(w|c_j) = \frac{\lambda + N_{c_j,w}}{\lambda |V| + \sum_{v=1}^{|V|} N_{c_j,v}} \quad (1)$$

where $N_{c_j,w}$ is the frequency of word w in documents of class c_j. $|V|$ is the size of vocabulary V and λ $(0 \leq \lambda \leq 1)$ is used for smoothing.

3.2 Components in LSC

This subsection describes our proposed method corresponding to the proposed LL components.

1. Past Information Store (PIS): In this work, we do not store the original data used in the past learning tasks, but only their results. For each past learning task $\hat{t}$, we store a) $P^{\hat{t}}(w|+)$ and $P^{\hat{t}}(w|-)$ for each word w which are from task $\hat{t}$'s NB classifier (see Eq 1); and b) the number of times that w appears in a positive $(+)$ document $N^{\hat{t}}_{+,w}$ and the number of times that w appears in a negative documents $N^{\hat{t}}_{-,w}$.
2. Knowledge Base (KB): Our knowledge base contains two types of knowledge:
   (a) Document-level knowledge $N^{KB}_{+,w}$ (and $N^{KB}_{-,w}$): number of occurrences of w in the documents of the positive (and negative) class in the past tasks, i.e., $N^{KB}_{+,w} = \sum_{\hat{t}} N^{\hat{t}}_{+,w}$ and $N^{KB}_{-,w} = \sum_{\hat{t}} N^{\hat{t}}_{-,w}$.
   (b) Domain-level knowledge $M^{KB}_{+,w}$ (and $M^{KB}_{-,w}$): number of past tasks in which $P(w|+) > P(w|-)$ (and $P(w|+) < P(w|-)$).
3. Knowledge Miner (KM). Knowledge miner is straightforward as it just performs counting and aggregation of information in PIS to generate knowledge (see 2(a) and 2(b) above).
4. Knowledge-Based Learner (KBL): This learner incorporates knowledge using regularization as penalty terms in our optimization. See the details in 3.4.

3.3 Objective Function

In this subsection, we introduce the objective function used in our method. The key parameters that affect NB classification results are $P(w|c_j)$ which are computed using empirical counts of word w with class c_j, i.e., $N_{c_j,w}$ (Eq. 1). In binary classification, they are $N_{+,w}$ and $N_{-,w}$. This suggests that we can revise these counts appropriately to improve classification. In our optimization, we denote the optimized variables $X_{+,w}$ and $X_{-,w}$ as the number of times that a word w appears in the positive and negative class. We called them *virtual counts* to distinguish them from empirical counts $N_{+,w}$ and $N_{-,w}$. For correct classification, ideally, we should have the posterior probability $P(c_j|d_i) = 1$ for labeled class c_j, and for the other class c_f, we should have $P(c_f|d_i) = 0$. Formally, given a new domain training data D^t, our objective function is:

$$\sum_{i=1}^{|D^t|} \left(P(c_j|d_i) - P(c_f|d_i)\right) \quad (2)$$

Here c_j is the actual labeled class of $d_i \in D^t$. In this paper, we use stochastic gradient descent (SGD) to optimize on the classification of each document $d_i \in D^t$. Due to the space limit, we only show the optimization process for a positive document (the process for a negative document is similar). The objective function under SGD for a positive document is:

$$F_{+,i} = P(+|d_i) - P(-|d_i) \quad (3)$$

To further save space, we omit the derivation steps and give the final derivatives below (See the detailed derivation steps in the separate supplementary note):

$$g(\boldsymbol{X}) = \left(\frac{\lambda |V| + \sum_{v=1}^{|V|} X_{+,v}}{\lambda |V| + \sum_{v=1}^{|V|} X_{-,v}}\right)^{|d_i|} \quad (4)$$

$$\frac{\partial F_{+,i}}{\partial X_{+,u}} = \frac{\frac{n_{u,d_i}}{\lambda + X_{+,u}} + \frac{P(-)}{P(+)} \prod_{w \in d_i} \left(\frac{\lambda + X_{-,w}}{\lambda + X_{+,w}}\right)^{n_{w,d_i}} \times \frac{\partial g}{\partial X_{+,u}}}{1 + \frac{P(-)}{P(+)} \prod_{w \in d_i} \left(\frac{\lambda + X_{-,w}}{\lambda + X_{+,w}}\right)^{n_{w,d_i}} \times g(\boldsymbol{X})} - \frac{n_{u,d_i}}{\lambda + X_{+,u}} \quad (5)$$

$$\frac{\partial F_{+,i}}{\partial X_{-,u}} = \frac{\frac{n_{u,d_i}}{\lambda + X_{-,u}} \times g(\boldsymbol{X}) + \frac{\partial g}{\partial X_{-,u}}}{\frac{P(+)}{P(-)} \prod_{w \in d_i} \left(\frac{\lambda + X_{+,w}}{\lambda + X_{-,w}}\right)^{n_{w,d_i}} + g(\boldsymbol{X})} \quad (6)$$

Alarm Clock	30.51	Flashlight	11.69	Home Theater System	28.84	Projector	20.24
Baby	16.45	GPS	19.50	Jewelry	12.21	Rice Cooker	18.64
Bag	11.97	Gloves	13.76	Keyboard	22.66	Sandal	12.11
Cable Modem	12.53	Graphics Card	14.58	Magazine Subscriptions	26.88	Vacuum	22.07
Dumbbell	16.04	Headphone	20.99	Movies TV	10.86	Video Games	20.93

Table 1: Names of the 20 product domains and the proportion of negative reviews in each domain.

where n_{u,d_i} is the term frequency of word u in document d_i. $\boldsymbol{X}$ denotes all the variables consisting of $X_{+,w}$ and $X_{-,w}$ for each word w. The partial derivatives for a word u, i.e., $\frac{\partial g}{\partial X_{+,u}}$ and $\frac{\partial g}{\partial X_{-,u}}$, are quite straightforward and thus not shown here. $X^0_{+,w} = N^t_{+,w} + N^{KB}_{+,w}$ and $X^0_{-,w} = N^t_{-,w} + N^{KB}_{-,w}$ are served as a reasonable starting point for SGD, where $N^t_{+,w}$ and $N^t_{-,w}$ are the empirical counts of word w and classes $+$ and $-$ from domain D^t, and $N^{KB}_{+,w}$ and $N^{KB}_{-,w}$ are from knowledge KB (Section 3.2). The SGD runs iteratively using the following rules for the positive document d_i until convergence, i.e., when the difference of Eq. 2 for two consecutive iterations is less than $1e-3$ (same for the negative document), where γ is the learning rate:

$$X^l_{+,u} = X^{l-1}_{+,u} - \gamma \frac{\partial F_{+,i}}{\partial X_{+,u}}, X^l_{-,u} = X^{l-1}_{-,u} - \gamma \frac{\partial F_{+,i}}{\partial X_{-,u}}$$

3.4 Exploiting Knowledge via Penalty Terms

The above optimization is able to update the virtual counts for a better classification in the target domain. However, it does not deal with the issue of domain dependent sentiment words, i.e., some words may change the polarity across different domains. Nor does it utilize the domain-level knowledge in the knowledge base KB (Section 3.2). We thus propose to add penalty terms into the optimization to accomplish these.

The intuition here is that if a word w can distinguish classes very well from the target domain training data, we should rely more on the target domain training data in computing counts related to w. So we define a set of words V_T that consists of distinguishable target domain dependent words. A word w belongs to V_T if $P(w|+)$ is much larger or much smaller than $P(w|-)$ in the target domain, i.e., $\frac{P(w|+)}{P(w|-)} \geq \sigma$ or $\frac{P(w|-)}{P(w|+)} \geq \sigma$, where σ is a parameter. Such words are already effective in classification for the target domain, so the virtual counts in optimization should follow the empirical counts ($N^t_{+,w}$ and $N^t_{-,w}$) in the target domain, which are reflected in the L2 regularization penalty term below (α is the regularization coefficient):

$$\frac{1}{2}\alpha \sum_{w \in V_T} \left(\left(X_{+,w} - N^t_{+,w}\right)^2 + \left(X_{-,w} - N^t_{-,w}\right)^2 \right) \quad (7)$$

To leverage domain-level knowledge (the second type of knowledge in KB in Section 3.2), we want to utilize only those reliable parts of knowledge. The rationale here is that if a word only appears in one or two past domains, the knowledge associated with it is probably not reliable or it is highly specific to those domains. Based on it, we use domain frequency to define the reliability of the domain-level knowledge. For w, if $M^{KB}_{+,w} \geq \tau$ or $M^{KB}_{-,w} \geq \tau$ (τ is a parameter), we regard it as appearing in a reasonable number of domains, making its knowledge reliable. We denote the set of such words as V_S. Then we add the second penalty term as follows:

$$\begin{aligned} &\frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{+,w} - R_w \times X^0_{+,w}\right)^2 \\ &+ \frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{-,w} - (1 - R_w) \times X^0_{-,w}\right)^2 \end{aligned} \quad (8)$$

where the ratio R_w is defined as $M^{KB}_{+,w}/(M^{KB}_{+,w} + M^{KB}_{-,w})$. $X^0_{+,w}$ and $X^0_{-,w}$ are the starting points for SGD (Section 3.3). Finally, we revise the partial derivatives in Eqs. 4-6 by adding the corresponding partial derivatives of Eqs. 7 and 8 to them.

4 Experiments

Datasets. We created a large corpus containing reviews from 20 types of diverse products or domains crawled from Amazon.com (i.e., 20 datasets). The names of product domains are listed in Table 1. Each domain contains 1,000 reviews. Following the existing work of other researchers (Blitzer et al., 2007, Pang et al., 2002), we treat reviews with rating > 3 as positive and reviews with rating < 3 as negative. The datasets are publically available at the authors websites.

Natural class distribution: We keep the natural (or skewed) distribution of the positive and negative reviews to experiment with the real-life situation. F1-score is used due to the imbalance.

NB-T	NB-S	NB-ST	SVM-T	SVM-S	SVM-ST	CLF	**LSC**
56.21	57.04	60.61	57.82	57.64	61.05	12.87	**67.00**

Table 2: Natural class distribution: Average F1-score of the negative class over 20 domains. Negative class is the minority class and thus harder to classify.

NB-T	NB-S	NB-ST	SVM-T	SVM-S	SVM-ST	CLF	**LSC**
80.15	77.35	80.85	78.45	78.20	79.40	80.49	**83.34**

Table 3: Balanced class distribution: Average accuracy over 20 domains for each system.

Balanced class distribution: We also created a balance dataset with 200 reviews (100 positive and 100 negative) in each domain dataset. This set is smaller because of the small number of negative reviews in each domain. Accuracy is used for evaluation in this balanced setting.

We used unigram features with no feature selection in classification. We followed (Pang et al., 2002) to deal with negation words. For evaluation, each domain is treated as the target domain with the rest 19 domains as the past domains. All the models are evaluated using 5-fold cross validation.

Baselines. We compare our proposed LSC model with Naïve Bayes (NB), SVM[1], and CLF (Li and Zong, 2008). Note that NB and SVM can only work on a single domain data. To have a comprehensive comparison, they are fed with three types of training data:

a) labeled training data from the target domain only, denoted by NB-T and SVM-T;
b) labeled training data from all past source domains only, denoted by NB-S and SVM-S;
c) merged (labeled) training data from all past domains and the target domain, referred to as NB-ST and SVM-ST.

For LSC, we empirically set $\sigma = 6$ and $\tau = 6$. The learning rate λ and regularization coefficient α are set to 0.1 empirically. λ is set to 1 for (Laplace) smoothing.

Table 2 shows the average F1-scores for the negative class in the natural class distribution, and Table 3 shows the average accuracies in the balanced class distribution. We can clearly see that our proposed model LSC achieves the best performance in both cases. In general, NB-S (and SVM-S) are worse than NB-T (and SVM-T), both of which are worse than NB-ST (and SVM-ST). This shows that simply merging both past domains and the target domain data is slightly beneficial. Note

[1]http://www.csie.ntu.edu.tw/˜cjlin/libsvm/

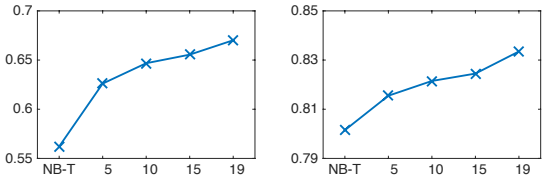

Figure 1: (Left): Negative class F1-score of LSC with #past domains in natural class distribution. (Right): Accuracy of LSC with #past domains in balanced class distribution.

that the average F1-score for the positive class is not shown as all classifiers perform very well because the positive class is the majority class (while our model performs slightly better than the baselines). The improvements of the proposed LSC model over all baselines in both cases are statistically significant using paired t-test ($p < 0.01$ compared to NB-ST and CLF, $p < 0.0001$ compared to the others). In the balanced class setting (Table 3), CLF performs better than NB-T and SVM-T, which is consistent with the results in (Li and Zong, 2008). However, it is still worse than our LSC model.

Effects of #Past Domains. Figure 1 shows the effects of our model using different number of past domains. We clearly see that LSC performs better with more past domains, showing it indeed has the ability to accumulate knowledge and use the knowledge to build better classifiers.

5 Conclusions

In this paper, we proposed a lifelong learning approach to sentiment classification using optimization, which is based on stochastic gradient descent in the framework of Bayesian probabilities. Penalty terms are introduced to effectively exploit the knowledge gained from past learning. Our experimental results using 20 diverse product review domains demonstrate the effectiveness of the method. We believe that lifelong learning is a promising direction for building better classifiers.

References

Alina Andreevskaia and Sabine Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *ACL*, pages 290–298.

Anthony Aue and Michael Gamon. 2005. Customizing Sentiment Classifiers to New Domains: A Case Study. In *RANLP*.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*, pages 440–447.

Danushka Bollegala, David J Weir, and John Carroll. 2011. Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification. In *ACL HLT*, pages 132–141.

Léon Bottou. 1998. Online algorithms and stochastic approximations. In David Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK. Oct 2012.

Andrew Carlson, Justin Betteridge, and Bryan Kisiel. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, pages 1306–1313.

Rich Caruana. 1997. Multitask Learning. *Machine learning*, 28(1):41–75.

Zhiyuan Chen and Bing Liu. 2014a. Mining Topics in Documents : Standing on the Shoulders of Big Data. In *KDD*, pages 1116–1125.

Zhiyuan Chen and Bing Liu. 2014b. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In *ICML*, pages 703–711.

Jianhui Chen, Jiayu Zhou, and Jieping Ye. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, pages 42–50.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification. In *ACL*, pages 123–131.

James J Heckman. 1979. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161.

Jing Jiang. 2008. A literature survey on domain adaptation of statistical classifiers. Technical report.

Lun-Wei Ku, Ting-Hao Huang, and Hsin-Hsi Chen. 2009. Using morphological and syntactic structures for Chinese opinion analysis. In *EMNLP*, pages 1260–1269.

Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *ACL HLT*, pages 257–260.

Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain Co-extraction of Sentiment and Topic Lexicons. In *ACL*, pages 410–419.

Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *AAAI*, pages 2127–2133.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.

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*, pages 79–86.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. Self-taught Learning : Transfer Learning from Unlabeled Data. In *ICML*, pages 759–766.

Paul Ruvolo and Eric Eaton. 2013. ELLA: An efficient lifelong learning algorithm. In *ICML*, pages 507–515.

Avishek Saha, Piyush Rai, Suresh Venkatasubramanian, and Hal Daume. 2011. Online learning of multiple tasks and their relationships. In *AISTATS*, pages 643–651.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, pages 49–55.

Songbo Tan, Gaowei Wu, Huifeng Tang, and Xueqi Cheng. 2007. A novel scheme for domain-transfer problem in the context of sentiment analysis. In *CIKM*, pages 979–982.

Sebastian Thrun. 1998. Lifelong Learning Algorithms. In S Thrun and L Pratt, editors, *Learning To Learn*, pages 181–209. Kluwer Academic Publishers.

Qiong Wu, Songbo Tan, and Xueqi Cheng. 2009. Graph Ranking for Sentiment Transfer. In *ACL-IJCNLP*, pages 317–320.

Rui Xia and Chengqing Zong. 2011. A POS-based Ensemble Model for Cross-domain Sentiment Classification. In *IJCNLP*, pages 614–622. Citeseer.

Yasuhisa Yoshida, Tsutomu Hirao, Tomoharu Iwata, Masaaki Nagata, and Yuji Matsumoto. 2011. Transfer Learning for Multiple-Domain Sentiment Analysis-Identifying Domain Dependent/Independent Word Polarity. In *AAAI*, pages 1286–1291.

Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *ICML*, page 114. ACM.

Jian Zhang, Zoubin Ghahramani, and Yiming Yang. 2008. Flexible latent variable models for multi-task learning. *Machine Learning*, 73(3):221–242.