Positive-Unlabeled Learning for Sentiment Analysis with Adversarial Training

Yueshen Xu[1,2], Lei Li[1], Jianbin Huang[1], Yuyu Yin[3], Wei Shao[4], Zhida Mai[5], and Lei Hei[6]

[1] School of Computer Science and Technology, Xidian University, Xi'an, 710071, China
ysxu@xidian.edu.cn, lli_3@stu.xidian.edu.cn,
jbhuang@xidian.edu.cn
[2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, Jiangsu, 215006, China
[3] School of Computer, Hangzhou Dianzi University, Hangzhou, 310018, China
yyy718@gmail.com
[4] School of Science, RMIT University, Melbourne, VIC 3001, Australia
wei.shao@rmit.edu.au
[5] Xanten Guangdong Development Co., Ltd, Foshan, 528200, China
top.mark@e-live.cn
[6] Center of Journal Publication, Xidian University, Xi'an, Shaanxi, 710071, China
heilei@xidian.edu.cn

Abstract. Sentiment classification is a critical task in sentiment analysis and other text mining applications. As a sub-problem of sentiment classification, positive and unlabeled learning or positive-unlabeled learning (PU learning) problem widely exists in real-world cases, but it has not been given enough attention. In this paper, we aim to solve PU learning problem under the framework of adversarial training and neural network. We propose a novel model for PU learning problem, which is based on adversarial training and attention-based long short-term memory (LSTM) network. In our model, we design a new adversarial training technique. We conducted extensive experiments on two real-world datasets. The experimental results demonstrate that our proposed model outperforms the compared methods, including the well-known traditional methods and state-of-the-art methods. We also report the training time, and discuss the sensitivity of our model to parameters.

Keywords: Sentiment Analysis · PU Learning Problem · Adversarial Training · LSTM · Attention Mechanism

1 Introduction

Sentiment analysis is one of key tasks in natural language processing (NLP) and has received a lot of attention in recent years [21, 26]. As a basic problem of sentiment analysis, *sentiment classification* aims to classify reviews or comments into different sentimental polarities [16]. Along with the rapid development of e-commerce sites and social networking sites, the volumes of reviews and comments increase dramatically,

and those online sites are in need of analyzing the polarities from large volumes of reviews and comments with high accuracy [5, 6].

Many works have tried to address sentiment classification using various techniques, including K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes, neural networks and some other methods [3, 29]. Meanwhile, as an effective technique for improving the robustness of machine learning methods, *adversarial training* has been also studied in tasks of text mining and natural language processing. In this paper, we exploit adversarial training in sentiment analysis, and there are few works exploiting adversarial training in sentiment analysis. For adversarial training, we have the following observations.

1. Traditional classifiers are likely to suffer from overfitting problem [8]. That is, a classifier overwhelmingly fits a certain words distribution in training reviews set and is trained to obtain a collection of parameters, but fails to fit the words distribution in test reviews set or new reviews set.
2. Adversarial examples are inputs formed by adding small perturbations with the intent of causing classifiers (e.g., neural networks) to misclassify [23], and can attack the generalization and fitness of classifiers. It can be inferred that a new classifier that is capable of resisting the attack of adversarial examples can achieve promising performance.

In this paper, we aim to solve the positive and unlabeled learning (PU learning) problem. In PU learning problem, there only exist positive labeled and unlabeled reviews, without any negative labeled reviews. PU learning problem indeed exists in real-world cases, and takes an important role in sentiment analysis [17]. PU learning problem was studied by previous works [13, 25]. In real-world cases, e-commerce sites and social networking sites can indeed confront PU learning problem. In this paper, we aim to solve PU learning problem based on adversarial training and neural network. Note that, there are several obstacles to exploit adversarial training in PU learning problem, including

1. Adversarial training requires that all training data have been labeled, but there are no negative labeled reviews in PU learning problem.
2. The evaluation metric for PU learning problem is usually F1-score, precision or recall, which cannot be modeled in the current loss function of adversarial training, which increases the difficulty in optimization during training process.

In this paper, we aim to solve PU learning problem comprehensively under the framework of adversarial training and neural network. In the proposed solution, we first identify negative reviews from unlabeled reviews. Then, we build an attention-based LSTM (Long Short-Term Memory) network, enhanced with an improved adversarial training method. We evaluate our models in two real-world datasets. The experimental results demonstrate that our models achieve the best performance, compared to a series of existing methods.

In summary, the contributions of this paper are as follows.

1. We propose a comprehensive solution to solve PU learning problem. The proposed solution is based on adversarial training and attentive LSTM network.

2. In PU learning problem, the procedure that distinguishes negative labels from unlabeled texts, is likely to introduce noise into training review texts, which can further hurt the classification performance. To tackle this issue, we propose an enhanced adversarial training method, adding a new perturbation to the word embeddings in LSTM network.
3. We conduct comparison experiments in two real-world datasets, and the experimental results demonstrate that our models can achieve better performance than compared methods. We also conduct sensitivity experiment to give instructions for selecting optimal parameter.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 elaborates the detail of our proposed model. Section 4 presents the experimental results and gives further analysis. Section 5 reports the sensitivity experimental result. Section 6 concludes the whole paper.

2 Related work

There have been many works studying sentiment classification, which employ a variety of methods, including machine learning methods (e.g., KNN, SVM and Naïve Bayes) and neural network methods [15]. PU learning (positive and unlabeled learning) problem is a sub-problem of sentiment classification, where there are no negative labels in corpus, but only positive and unlabeled reviews. PU learning problem also exists in real-world e-commerce sites and social networking sites. As for adversarial training, it has been verified to be effective to improve the robustness of models in many applications.

PU learning problem was first studied in [17]. In PU learning problem, there is a preliminary task to construct a classifier to identify negative labeled reviews from positive and unlabeled review texts. [14] proposed a whole framework and identified negative labeled reviews using Rocchio technique. [4] studied the design of the loss function in PU learning problem. The authors established the generalization error bounds for loss function in PU learning problem. [25] and [13] focused on a specific but valuable problem, that was, to detect deceptive reviews from positive and unlabeled reviews. In this paper, we aim to solve PU learning problem based on adversarial training and attentive neural network. The procedure of identifying negative labeled reviews is highly likely to bring erroneous labels, and we design a new adversarial training method to attack these erroneous labels, further improving the classification performance.

Adversarial training aims to improve the robustness of machine learning models by exposing a model to adversarial examples during training process. Adversarial training was first introduced in the problem of image classification, where the input image pixels were continuous values [8], and researchers studied adversarial training technique from many aspects [12, 7]. [23] proposed a new algorithm of crafting adversarial examples. [20] adapted adversarial training to solve text classification in a semi-supervised setting. [27] employed adversarial training in relation extraction problem and proposed an improved neural network architecture. As a prevailing tool in many artificial intelligence tasks, the generative adversarial net (GAN) proposed by [7] also borrows ideas from adversarial training. In this paper, we exploit the potential of adversarial training in improving the performance of PU learning problem.

3 Adversarial training for sentiment classification

In this section, we first state the base models employed in our methods, and then elaborate the proposed model for PU learning problem.

3.1 The base models

LSTM network. Recurrent neural network (RNN) takes sequential data as input, and finishes the computation via recursive cells. Standard RNN has several problems in training process, such as gradient vanishing and gradient exploding. To address these issues, LSTM network is developed and achieves superior performance [9]. Formally, each cell in LSTM is computed as follows.

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{1}$$

$$f_t = \sigma(W_f \cdot X + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c) \tag{4}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

At time step t, the previous hidden output h_{t-1} and the current input x_t together form the input X (see Eq. 1). There are three gates in an LSTM cell, which are forget gate, input gate and output gate. Forget gate outputs a value in $[0, 1]$, to indicate the amount of information from previous cell that need to be dumped in Eq. 2. Input gate first decides those values that LSTM will update by i_t in Eq. 3, and further computes a candidate cell state c_t using Eq. 4. Finally, the output gate decides which part of the candidate cell states will be outputted, and the cell output h_t is computed by Eq. 6.

Let T be a piece of review represented by a sequence of m words, as $T = \{w_t | t = 1, \ldots, m\}$, and T is tagged with a label as y. Each word w_t is embedded into a k-dimensional word vector $v_t = \mathbf{W} \times w_t$, where $\mathbf{W} \in \mathbb{R}^{k \times |V|}$ is a word embedding matrix to be learned, and V denotes the vocabulary. Figure 1 shows the basic LSTM model for classification task in NLP. w_{eos} denotes the end mark of a review, and v_{eos} is the word embedding result of w_{eos}.

Attention mechanism. In recent years, attention mechanism has become a compelling technique in sequence models, which can improve the capability of models in handling long-range dependencies [2]. In NLP tasks, attention mechanism gives the model a chance to capture the important part of the input that needs more attention. Guided by a weight vector learned from the input text and the result that is produced so far, attention-based model captures more information based on a more comprehensive

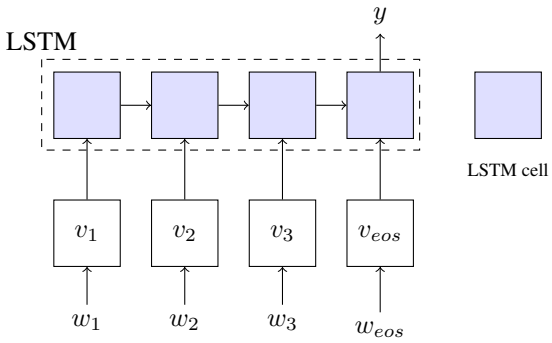

Fig. 1. The basic LSTM model for classification task in NLP

modeling of the input. In detail, we learn an attention vector $\boldsymbol{\alpha}$ as follows.

$$u_i = tanh(W_s h_i + b_s), \tag{7}$$

$$\alpha_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)}, \tag{8}$$

$$\omega = \sum_i \alpha_i h_i \tag{9}$$

where α_i denotes each element in $\boldsymbol{\alpha}$. The final output of an attentive LSTM is ω (see Eq. 9), which can be treated as a weighted sum over all outputs of all cells in LSTM. With the output ω of the attentive LSTM network, a fully connected layer and a softmax non-linear layer are used to map ω to the probability distribution over each class, and further to obtain the label y.

3.2 The proposed model for PU learning problem

Compared to traditional sentiment classification, in PU learning problem, there is one extra step before conducting classification. The step is to distinguish the reviews or comments with negative labels from the positive and unlabeled review texts. We adopt a two-step strategy to finish this task, following the suggestions in [17]. In detail, we use the Rocchio technique [19] to generate positive and potential negative review texts from unlabeled review texts. In Rocchio technique, each document is represented by a vector, and each element in the vector is the value that is computed with tf-idf (term frequency-inverse document frequency).

Let D denote the whole set of training texts, and let C_j denote the set of training reviews in class c_j. In this paper, we have two classes, that is, j being 1 represents the positive review and j being -1 represents the negative review. To build a Rocchio classifier, a representative vector $\boldsymbol{c}_j$ of C_j is constructed for each class c_j first, which is as follows.

$$\boldsymbol{c}_j = \eta \frac{1}{|C_j|} \sum_{\boldsymbol{d} \in C_j} \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} - \rho \frac{1}{|D - C_j|} \sum_{\boldsymbol{d} \in D - C_j} \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} \tag{10}$$

where η and ρ are parameters that control the weights of similar and dissimilar training examples. $\boldsymbol{d}$ denotes a piece of review, and $\|\boldsymbol{d}\|$ denotes the norm of review $\boldsymbol{d}$ (i.e., the number of words in $\boldsymbol{d}$). Then for each test review td, the similarity of td with each representative vector is measured by cosine similarity. Finally, td is assigned to the class, the representative vector of which is the most similar to td. We use P to denote the positive set and U to denote the unlabeled set. The overall procedure of Rocchio technique is stated as follows.

1. Assign each review in P to class label 1;
2. Assign each review in U to class label -1;
3. Build a Rocchio classifier using P and U;
4. Use the classifier to classify U. Those reviews in U that are classified to be negative will form the negative reviews set.

Although the dataset is fully formed, the potential unreliability of the identified negative labeled texts increases the noise that is likely to harm the performance of neural networks. More specifically, the noise refers to the positive reviews which are labeled to be negative by the Rocchio classifier. To tackle those noise data in PU learning problem, we add a new random perturbation $\boldsymbol{r}$ to word embedding results E, due to the following two reasons.

1. The first is that adding a new random perturbation can help the gradient computation escape from the non-smooth surrounding area of each word embedding [12].
2. The second is that a random perturbation on word embeddings input can take the role of regularization to defend the potential overfitting.

The added random perturbation in current word embedding results E generates new perturbed word embedding results E', which are as follows.

$$\boldsymbol{r} = \beta \times sign(\mathcal{N}(\mathbf{0}^k, \mathbf{I}^k)) \tag{11}$$

$$E' = E + \boldsymbol{r} \tag{12}$$

$$\boldsymbol{e}_{adv} = \epsilon \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}, \text{ where } \boldsymbol{g} = \nabla_{E'} L(E'; \hat{\Theta}) \tag{13}$$

We choose Gaussian distribution to generate the random adversarial perturbation (Eq. 11). $\mathcal{N}(\mathbf{0}^k, \mathbf{I}^k)$ is the Gaussian distribution, where $\mathbf{0}^k$ is the mean vector and $\mathbf{I}^k$ is the covariance matrix (k is the dimension of word embedding). β controls the extent of trusting Gaussian distribution to generate the adversarial perturbation. $sign(\cdot)$ is the multi-dimensional indicator function, and the input of $sign(\cdot)$ is a k dimensional vector. The loss function for PU learning problem is constructed as follows.

$$L_{cls} = L(E'; \Theta) \tag{14}$$

$$L_{adv} = L(E' + \boldsymbol{e}_{adv}; \Theta) \tag{15}$$

$$\hat{L}(\Theta) = \alpha L_{cls} + (1 - \alpha) L_{adv} \tag{16}$$

where $\boldsymbol{e}_{adv}$ and Θ are two parameters in adversarial training. α is a parameter to control the ratio of classification loss and adversarial loss. We name the proposed model as *PU*

learning problem with Adversarial Training (PUAT for short). Figure 2 demonstrates the model of PUAT, where $r^{(i)}$ $(i = 1, 2, \ldots)$ denotes the element in $\boldsymbol{r}$ and $e^{(i)}$ $(i = 1, 2, \ldots)$ denotes the element in $\boldsymbol{e}_{adv}$. h_i $(i = 1, 2, \ldots)$ is the hidden output of each LSTM cell.

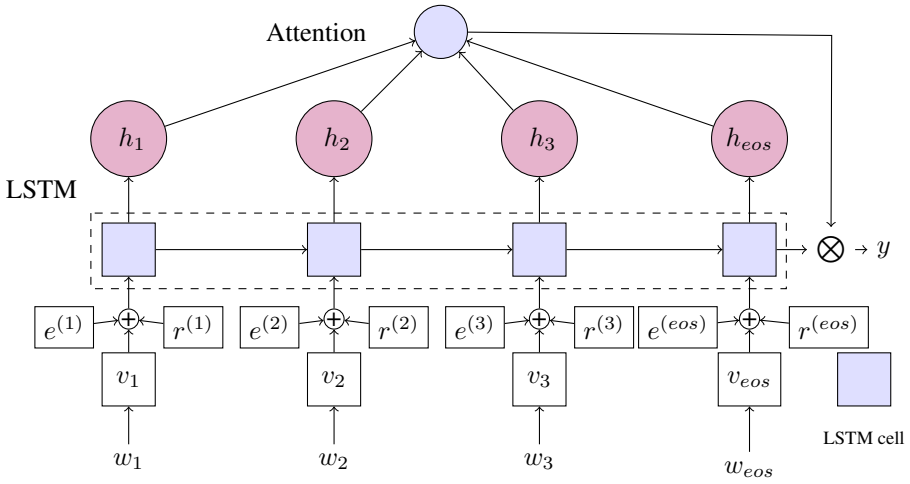

Fig. 2. The PUAT model with perturbed and random word embeddings and attentive LSTM

4 Experiment and Evaluation

4.1 Experimental setting

Datasets.We evaluated our methods on two real-world datasets, i.e. IMDB dataset and Elec dataset. The IMDB dataset contains movie reviews and has been widely used in evaluation of sentiment classification [18]. The Elec dataset contains electronic product reviews collected from Amazon and has been also widely used in sentiment classification tasks [10]. The statistics of the two datasets are shown in Table 1.

Table 1. Statistics of IMDB dataset and Elec dataset

Dataset	*#label*	*#training*	*#test*	*Avg.*	*Max*
IMDB dataset	2	25000	25000	239	2506
Elec dataset	2	25000	25000	107	4983

In Table 1, *#label* denotes the number of classes, and there are two sentiment classes in both datasets, including positive class and negative class. *#training* and *#test* represent the number of reviews in the training set and test set. *Avg.* is the average length

of reviews in each dataset, and *Max* denotes the maximum length of the reviews. We randomly selected 90% reviews from training reviews set to form the training set and the remained 10% reviews form the validation set.

Implementation. We implemented our codes in TensorFlow [1]. We compare our models to the following models that can be used to solve the classification problem on review texts, including

1. SVM and Naïve Bayes. SVM and Naïve Bayes are two classic classification methods that are also commonly used in text mining and natural language processing applications. [22] showed that the two methods can be used in sentiment classification problem.
2. LSTM (Long-Short Term Memory) and GRU (Gated Recurrent Unit). LSTM and GRU are two popular variants of RNN, and have been also applied on sentiment classification tasks. We built an LSTM network with the hidden size being 128. The parameters and configuration of GRU network are the same as those in LSTM.
3. Attention-based LSTM or attentive LSTM. This model is proposed in [28], as a hierarchical attention-based LSTM network, and achieves good performance on a series of text classification tasks.
4. Adversarial LSTM. This model is proposed in [20], which adapts adversarial training in basic LSTM network and achieves the state-of-the-art performance on semi-supervised classification problem.

Regarding the preprocessing, the words whose document frequencies are less than 2 are removed from the reviews in both datasets. The reason is that, those words that less frequently appear will enlarge the whole vocabulary size, and further obviously increase training time. In our proposed model PUAT, the LSTM network is configured with 128 hidden units.

Parameter Setting. The parameters are set based on the evaluation results on the validation set. The word embeddings are initialized by GloVe [24], and the dimension of word embedding vector k is 200. The parameter α in Eq. 16 is set to 0.5. The parameter ϵ in Eq. 13 is set to 1.0.

For the optimization of model parameters, we used Adam optimizer [11]. Based on the results on validation set, we set the learning rate to 0.001, batch size to 256 and dropout rate to 0.8. The parameters of the compared methods are set according to the default settings in referred papers.

4.2 The generation of training sets in PU learning problem

Similar to [17], we take the following steps to generate PU learning problem dataset from the training set. We randomly select p percent of positive reviews as the positive set P, and the remaining positive reviews and negative reviews are disassociated with their labels, and are used to form the unlabeled set U. The task is to classify negative reviews from unlabeled set U. We change the value of p in the range from 20% to 40% to provide a comprehensive evaluation.

As stated in Section 3.2, we used $tf\text{-}idf$ to compute the weight of each word and further to form feature vectors. We then built a Rocchio classifier on the positive set P

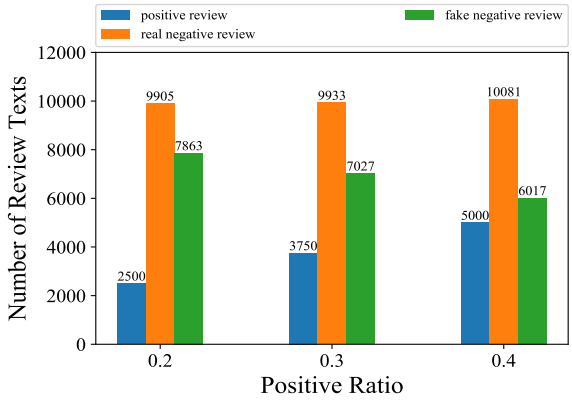

Fig. 3. The number of review texts of the three different review types in IMDB dataset. The bars from left to right represent the positive review, real negative review and fake negative review respectively.

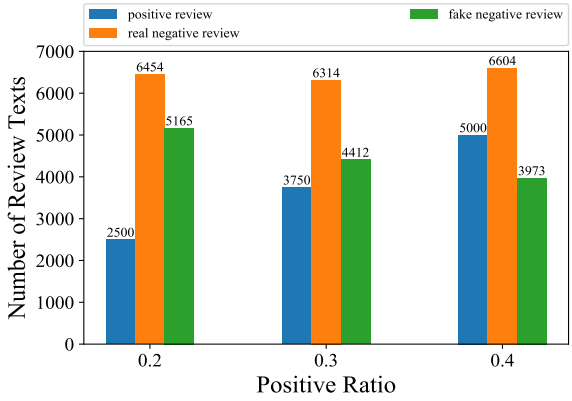

Fig. 4. The number of review texts of the three different review types in Elec dataset. The bars from left to right represent the positive review, real negative review and fake negative review respectively.

and unlabeled set U. The reviews in set U that are classified to be negative form the negative set N. Figure 3 (IMDB dataset) and Figure 4 (Elec dataset) show the details of negative reviews generated by Rocchio classifier with different positive ratios (0.2, 0.3 and 0.4). It can be seen that the whole training set is divided into positive set and negative set. The negative set consists of two parts, including real negative review texts and fake negative review texts. The fake negative review texts are the original positive reviews but misclassified as negative reviews by Rocchio classifier. Also, it can be found that along with the positive ratio increasing (0.2 to 0.4), the proportion of fake negative review texts decreases.

4.3 Experimental results in PU learning problem

We conduct the experiments under three cases of positive ratios (0.2, 0.3 and 0.4), which correspond to the positive ratio settings in Section 4.2. We will report the test performance of each method. The evaluation metrics include F1-score, recall and test accuracy. F1-score is a widely used metric in classification problem and tends to give an integrated evaluation, as F1-score combines recall and precision. The reported F1-score is computed on positive class, as in PU learning problem, there are only positive labeled reviews. F1-score is computed as

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{17}$$

To give a comprehensive evaluation, we also report the results of recall and test accuracy. Furthermore, we will discuss the relationship between the positive ratio and test performance. In the generation of training sets, we have generated positive labeled sets P and negative labeled sets N under different cases of positive ratios (0.2, 0.3 and 0.4). Different positive ratios decide different proportions of positive reviews and negative reviews in final training set.

Table 2. Test performance in IMDB dataset and Elec dataset in PU learning problem with positive ratio being 0.2.

Method	IMDB dataset			Elec dataset		
	F1-score	Recall	Test Accuracy	F1-score	Recall	Test Accuracy
Naïve Bayes	0.159	0.086	0.509	0.608	0.480	0.595
SVM	0.341	0.207	0.599	0.777	0.706	0.797
GRU	0.526	0.368	0.655	0.760	0.673	0.783
LSTM	0.512	0.354	0.647	0.772	0.706	0.779
Attentive LSTM	0.609	0.458	0.695	0.775	0.721	0.786
Adversarial LSTM	0.530	0.370	0.665	0.799	0.739	0.808
PUAT	**0.650**	**0.502**	**0.723**	**0.804**	**0.772**	**0.814**

Tables 2, 3 and 4 present F1-score, recall and test accuracy results of all methods in PU learning problem. It can be found that the proposed PUAT method achieves the highest F1-score, recall and test accuracy values in all three positive ratio settings (0.2, 0.3 and 0.4). Furthermore, we can have following observations.

1. First, in all three positive ratio settings, the proposed PUAT method achieves better F1-score and test accuracy results than the traditional classifiers, including SVM and Naïve Bayes. Take SVM as an example. In the case of positive ratio being 0.2, SVM achieves a 0.341 F1-score and a 0.599 test accuracy in IMDB dataset and a 0.777 F1-score and a 0.797 test accuracy in Elec dataset. In contrast, PUAT achieves a higher performance of a 0.650 F1-score and a 0.723 test accuracy in IMDB dataset and a 0.804 F1-score and a 0.814 test accuracy in Elec dataset, also in the case of positive ratio being 0.2.

Table 3. Test performance in IMDB dataset and Elec dataset in PU learning problem with positive ratio being 0.3.

Method	IMDB dataset			Elec dataset		
	F1-score	Recall	Test Accuracy	F1-score	Recall	Test Accuracy
Naïve Bayes	0.280	0.164	0.527	0.704	0.626	0.670
SVM	0.509	0.347	0.666	0.824	0.807	0.824
GRU	0.676	0.531	0.742	0.815	0.796	0.823
LSTM	0.683	0.539	0.747	0.820	0.805	0.825
Attentive LSTM	0.719	0.588	0.772	0.815	0.806	0.822
Adversarial LSTM	0.689	0.544	0.764	0.819	0.779	0.828
PUAT	**0.815**	**0.751**	**0.832**	**0.825**	**0.808**	**0.839**

Table 4. Test performance in IMDB dataset and Elec dataset in PU learning problem with positive ratio being 0.4.

Method	IMDB dataset			Elec dataset		
	F1-score	Recall	Test Accuracy	F1-score	Recall	Test Accuracy
Naïve Bayes	0.421	0.270	0.565	0.732	0.686	0.696
SVM	0.626	0.464	0.723	0.829	**0.831**	0.828
GRU	0.752	0.639	0.788	0.821	0.813	0.830
LSTM	0.774	0.675	0.792	0.825	0.803	0.832
Attentive LSTM	0.775	0.681	0.793	0.831	0.823	0.839
Adversarial LSTM	0.799	0.716	0.813	0.830	0.819	0.838
PUAT	**0.818**	**0.749**	**0.824**	**0.835**	**0.831**	**0.845**

2. PUAT also outperforms the state-of-the-art methods. For example, in the case of positive ratio being 0.3, adversarial LSTM achieves a 0.689 F1-score and a 0.764 test accuracy in IMDB dataset, and achieves a 0.819 F1-score and a 0.828 test accuracy in Elec dataset. In contrast, the proposed PUAT achieves a superior performance of a 0.815 F1-score and a 0.832 test accuracy in IMDB dataset, along with a 0.825 F1-score and a 0.839 test accuracy in Elec dataset.

3. Compared to the performance achieved by LSTM and GRU, the F1-score results of SVM are competitive, especially in Elec dataset. That is, the F1-score results of SVM are close to those of LSTM and GRU or even better than those of LSTM and GRU.

   An important reason is in the existence of misclassified reviews (noise), i.e., the true positive review texts that are misclassified to be negative. The LSTM network is easy to suffer from overfitting on such kind of noise. As for SVM, the positive set P and negative set U have been generated by Rocchio classifier, and such data separation provides a preliminary preparation for SVM to find a strong margin to separate positive and negative classes. The noise also leads to bad performances of Naïve Bayes, and Naïve Bayes tends to predict all reviews in test set to be negative.

4. For the analysis of recall, let us start from the computation of recall, which is given by
$$recall = \frac{\text{TP}}{\text{TP + FN}} \tag{18}$$
where TP (short for **true positive**) denotes the number of positive reviews that are correctly predicted as positive labeled reviews. FN (short for **false negative**) denotes the number of positive reviews that are misclassified to be negative reviews. In our model, adversarial training improves the ability of distinguishing true negative reviews from fake negative reviews, which decreases the number of fake negative reviews.
5. Besides, we can make some interesting observations on the positive ratio p. When p is equal to 1.0, PU learning problem turns to a case that all positive labeled reviews in training set are reserved. When p is near to 0, it indicates that there are few positive review texts, and it will be difficult to build an effective classifier purely using unlabeled review texts. From Tables 2, 3 and 4, it can be found that there is a positive correlation between the evaluation metrics and positive ratio p. The reason is that the increasing number of positive review texts can help the classifier learn more useful features.

4.4 Training time comparison

Training time is another concern when people deploy neural network models. In this section, we compare the training time of different models in IMDB dataset and Elec dataset. As neural network models are usually trained by iterations of epochs, we report the average training time of one epoch when positive ratio p is 0.4. All experiments were conducted on a machine equipped with a Xeon E5-2680 v4 CPU and a single NVIDIA Telsa M40 GPU. The compared results are present in Table 5.

Table 5. Training time comparison with positive ratio p being 0.4

Model	Average training time of one epoch	
	IMDB dataset	Elec dataset
Attentive LSTM	7.28s	6.43s
Adversarial LSTM	16.85s	15.06s
PUAT	19.38s	17.05s

From Table 5, it can be found that the models using adversarial training techniques (adversarial LSTM and PUAT) require more training time. The reason is that the models using adversarial training techniques require to compute the adversarial perturbations. As for our model, PUAT needs to compute the gradients also on E in Eq. 13.

Furthermore, it can be seen that the average training time of our models is slightly higher than that of adversarial LSTM model. Note that, our proposed models achieve better performances, which have been verified by the experimental results in previous sections. Take PUAT as an example. Compared to adversarial LSTM in IMDB dataset,

PUAT achieves higher F1-score, recall and test accuracy (see Table 4), while PUAT only spends 2.53s more in one epoch (19.38s for PUAT and 16.85s for adversarial LSTM). The results indicate that considering the superior performances, the training time of our proposed models is acceptable.

5 Impact of β

The parameter β controls the trust extent of the adversarial perturbation generated from Gaussian distribution (see Eq. 11 in Section 3.2). If β is equal to 0, the proposed PUAT model degenerates into the ordinary adversarial training method. If β is a large value, the random perturbations may exert too much impact on word embeddings, which probably further harms the result of word embedding. We study the sensitivity of PUAT model to β, and take the case that the positive ratio is equal to 0.3 as an example. The experimental results are shown in Figure 5, where the horizontal axis is set as the logarithmic coordinate ranging from 10^{-3} to 10^{1}.

It can be found that the optimal value of β is achieved around the value of 10^{0} (i.e., 1.0) in both datasets. The change trend of β in IMDB dataset is smoother than the change trend in Elec dataset. These observations mean that the value of β should be set not too large or too small. In our experiments, β is set to 1.0 in all experiments.

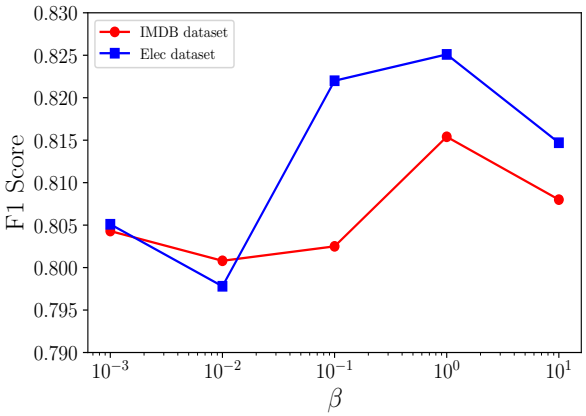

Fig. 5. F1-score varies with β when the positive ratio is equal to 0.3

6 Conclusion

In this paper, we give a comprehensive study of adversarial training in PU learning problem. The proposed model is built based on adversarial training and attentive LSTM network, and is named as PUAT (*PU learning with Adversarial Training*). To the best

of our knowledge, this is the first paper that conducts fully study of adversarial training in PU learning problem.

In two datasets, the experimental results demonstrate that our proposed models achieve superior performance than the compared models. Such superiority verifies the effectiveness of the proposed way of using attention mechanism and adversarial training in our model. We gave a detailed discuss on experimental results. We also discussed the parameter sensitivity and reported the comparison results of training time.

Acknowledgements

This paper is granted by Fundamental Research Fund for Central Universities (No. JBX171007), National Natural Science Fund of China (No.61702391), Natural Science Foundation of Shaanxi province and Zhejiang province (No.2018JQ6050, No. LY12F02003). Meanwhile, the authors would like to thank Minhao Ni for her valuable suggestions in experiment design.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., etc.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems (2015)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
3. Dey, L., Chakraborty, S., Biswas, A., Bose, B., Tiwari, S.: Sentiment analysis of review datasets using nave bayes and k-nn classifier. International Journal of Information Engineering and Electronic Business (4), 54–62 (2016)
4. Du Plessis, M.C., Niu, G., Sugiyama, M.: Analysis of learning from positive and unlabeled data. In: Advances in Neural Information Processing Systems (NIPS). pp. 703–711 (2014)
5. Gao, H., Huang, W., Yang, X., Duan, Y., Yin, Y.: Toward service selection for workflow reconfiguration: An interface-based computing solution. Future Generation Computer Systems **87**, 298–311 (2018)
6. Gao, H., Mao, S., Huang, W., Yang, X.: Applying probabilistic model checking to financial production risk evaluation and control: A case study of alibaba's yu'e bao. IEEE Transactions on Computational Social Systems **5**(3), 785–795 (2018)
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NIPS). pp. 2672–2680 (2014)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations (ICLR). pp. 1–11 (2015)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Johnson, R., Zhang, T.: Semi-supervised convolutional neural networks for text categorization via region embedding. In: Advances in Neural Information Processing Systems (NIPS). pp. 919–927 (2015)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)

12. Kurakin, A., Boneh, D., Tramr, F., Goodfellow, I., Papernot, N., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018)
13. Li, H., Liu, B., Mukherjee, A., Shao, J.: Spotting fake reviews using positive-unlabeled learning. Computacióny Sistemas **18**(3), 467–475 (2014)
14. Li, X.L., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: European Conference on Machine Learning (ECML). pp. 218–229. Springer (2005)
15. Lin, J., Mao, W., Zeng, D.D.: Personality-based refinement for sentiment classification in microblog. Knowledge-Based Systems **132**, 204–214 (2017)
16. Liu, B.: Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press (2015)
17. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Third IEEE International Conference on Data Mining (ICDM). pp. 179–186 (2003)
18. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL). pp. 142–150 (2011)
19. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press (2009)
20. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017)
21. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2), 1–135 (2007)
22. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). pp. 79–86 (2002)
23. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy (EuroS&P). pp. 372–387 (2016)
24. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543 (2014)
25. Ren, Y., Ji, D., Zhang, H.: Positive unlabeled learning for deceptive reviews detection. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 488–498 (2014)
26. Wang, Y., Huang, M., Zhao, L., Zhu, X.: Attention-based lstm for aspect-level sentiment classification. In: Proceedings of International Conference on Empirical Methods in Natural Language Processing (EMNLP) (2016)
27. Wu, Y., Bamman, D., Russell, S.: Adversarial training for relation extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1778–1783 (2017)
28. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of NAACL. pp. 1480–1489 (2016)
29. Yuan, Z., Wu, S., Wu, F., Liu, J., Huang, Y.: Domain attention model for multi-domain sentiment classification. Knowledge-Based Systems **155**, 1–10 (2018)