Sentiment Analysis via Dually-born-again Network and Sample Selection

Pinlong Zhao[a], Zefeng Han[a], Qing Yin[a], Shuxiao Li[b], and Ou Wu*[a]

[a] *Center for Applied Mathematics, Tianjin University, Tianjin, China*
[b] *Insituite of Automation, Chinese Academy of Sciences, Beijing, China*

Abstract

Text sentiment analysis is an important natural language processing (NLP) task and has received considerable attention in recent years. Numerous deep-learning based methods have been proposed in previous literature in terms of new deep neural networks (DNN) including new embedding strategies, new attention mechanisms, and new encoding layers. In this study, an alternative technical path is investigated to further improve the state-of-the-art performance of text sentiment analysis. An new effective learning framework is proposed that combines knowledge distillation and sample selection. A dually-born-again network (DBAN) is presented in which the teacher network and the student network are simultaneously trained through an iterative approach. A selection gate is defined to deal with training samples which are useless or even harmful for model training. Moreover, both the DBAN and sample selection are further improved by ensemble. The proposed framework can improve the existing state-of-the-art DNN models in sentiment analysis. Experimental results indicate that the proposed framework enhances the performances of existing networks. In addition, DBAN outperforms existing born-again network.

Keywords: Classification, deep neural network, knowledge distillation, sample selection

1 Introduction

Text sentiment analysis is a key component in various text mining applications [1, 2]. Its goal is to accurately classify given a text sample into different categories, which are usually set as three-level {positive, neural, negative} or five-level {very negative, negative, neutral, positive, very positive}. Deep neural network (DNN) has become an extensively used learning technique in sentiment analysis because it does not require hand-crafted features and has a remarkable

*Corresponding Author: Ou Wu, Center for Applied Mathematics, Tianjin University, Tianjin, China.
E-mail: pinlongzhao@tju.edu.cn, hanzefeng@tju.edu.cn, qingyin@tju.edu.cn, shuxiao.li@ia.ac.cn, wuou@tju.edu.cn

performance [3, 4, 5]. Numerous methods have been proposed in previous papers. These existing methods focus on the modification of network modules including new embedding [6], new attention [7], or new encoding layers [8, 9]. In addition, a few studies have attempted to leverage additional knowledge to further improve performance [10, 11]. Promising results have been obtained along these technical paths.

Rather than introducing new effective DNN models or utilizing extra domain knowledge, the deep leaning community also emerges some novel and effective techniques to improve classification performances of existing DNN models. One of such techniques is knowledge distillation, which refers to the distillation of knowledge from a trained teacher network to guide the training of a student network without modifying the network structures. Numerous applications [12] of distillation in computer vision demonstrate that knowledge distillation enhances the performance of a student network. Furlanello et al. investigated a special case of knowledge distillation [13], namely, born-again network (BAN), in which the teacher and the student networks share the same structure. BAN has been proven to be effective in solving various problems.

Apart from knowledge distillation, noisy-label learning is another extensively used learning strategy without keeping eyes on new learning models. Labels in some training data may contain errors due to label difficulty or annotators' carelessness [14]. Several studies have proposed solutions to deal with noisy labels [15]. This work brings the main idea of noisy-label learning for sample selection. The motivation for sample selection is that a small-proportion of training samples may play a negative role in the model training and these samples can be seen as noisy.

In addition, ensemble learning [16] combines a set of existing basic learning methods to produce a more effective model. Ensemble learning has been integrated in knowledge distillation [17] to provide valuable knowledge to supervise a student model.

Inspired by the above progress in machine-learning strategies, we propose a new learning framework rather than single learning model design for sentiment analysis in this paper. First, a new dually-born-again (DBAN) learning approach is proposed, in which the teacher network and the student network are trained simultaneously in each iteration. They share the same network structure and most parameters. Second, several existing sentiment analysis models are used and their outputs are combined to guide the learning of the teacher and student networks inspired by knowledge distillation. Further, a selection gate is defined to deal with training samples which are useless or even harmful for model training. Experiments on two benchmark data sets indicate that the proposed framework can improve the learning performance of the existing state-of-the-art DNN-based sentiment analysis models. Our work is innovative in the following aspects:

- A new learning framework is proposed to further improve the performance of existing DNN models for text sentiment analysis. This framework is inspired by the related studies on knowledge distillation, noisy-label learning, and ensemble learning.

- A new born-again learning approach is proposed. Our approach simultaneously trains the teacher and student networks with mixed supervised

signals. This approach is more effective and efficient than the existing born-again strategy.

- To further improve the quality of the training samples, ensemble outputs are used to construct a selection gate that can filter samples that are relatively difficult to learn or even harmful.

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation is initially designed for model compression [18, 19]. Knowledge of distillation usually refers to the distribution outputs of each training sample of a teacher model. This method borrows the knowledge from a cumbersome but high-performance model (teacher) to develop a simple model (student). The teacher model can be an ensemble of models and the student model can be a single model. Various experiments have shown that the class probabilities produced by a teacher model are better than the original ground-truth labels in training of a student model [20]. Knowledge distillation has been used in various natural language processing (NLP) tasks including neural machine translation [21]. To the best of our knowledge, knowledge distillation has not been applied to sentiment analysis.

Furlanello et al. (2018) proposed a new knowledge distillation strategy, in which the teacher and the student models share the same structure [13], to improve the performance of an existing DNN. They are optimized iteratively (teacher $\rightarrow$ student $\rightarrow$ teacher$\rightarrow$ student) while the next generation is guided by the standard ground-truth labels and the class probabilities in the previous generation. Extensive experiments have shown that the final student model outperforms the original teacher model.

2.2 Text Classification

Numerous text sentiment classification methods have been proposed in previous literature and can be divided into three categories. The first category is rule-based. Rule-based methods are also known as lexicon-based methods in which dictionary of three kinds (positive, negative, negation) of the key words are compiled. A set of rules are subsequently constructed based on the appearance and positional relationships between key words. The second category is (conventional) learning-based. One-hot word-level features are constructed and fed into a shallow learning model (e.g., **SVM** and **Adaboost**). The third category is deep learning-based.CNN and long short-term memory (LSTM) are often used to encode input texts and a softmax classifier is used to predict the sentiment category [3, 22]. Previous deep learning-based methods have focused on new network structures (wang et al. 2018), new attention mechanisms, or the utilization of domain knowledge. Our study adopts a new technical path to further improve the state-of-the-art sentiment analysis performance.

2.3 Noisy-label Learning

Noisy-label learning assumes that a small proportion of labels in training data are errors caused by labeling noise [14]. The majority of the noisy-label learning methods attempt to model the labeling noise and than infer the ground-truth labels[23]. Some other methods assume that there is an additional small-size training set with high-quality labels [24]. Knowledge distillation has been used in noisy-label learning [15]. This study attempts to leverage the ensemble learning to deal with samples that are difficult to train. These samples can be considered with noisy labels or ambiguous. Our study focus is not on learning with noise samples, but on improving the performance of the existing DNN models by mining more information of samples. The difference between our study and noisy-label learning methods is that noisy-label learning attempts to model the labeling noise and than infer the ground-truth labels, but our model attempts to improving the performance of the existing DNN models by finding out the difficult-to-train samples.

3 Methodology

Text sentiment analysis can be formulated as follows. Given a piece of input texts $s = \{x_1, \cdots, x_n\}$. The analysis goal is to predict the overall sentiment class of s. Therefore, a sentiment classifier is required to be constructed that maps s into a sentiment class in predefined sentiment categories.

3.1 The Overall Learning Framework

This study aims to design a learning framework to improve the performance of a mainstream existing DNN model. Accordingly, a dually-born-again network and sample selection are leveraged. DBAN can facilitate the performance of a single DNN model. Meanwhile, sample selection can select samples that are useless or even harmful for model. Both DBAN and sample selection can further improve by ensemble learning. Besides, no domain knowledge is used in the entire process. Let $\{X, Y\}$ be the training data. Assuming that there are three existing models (C1, C2, and C3) for ensemble. The entire architecture of our framework is shown in Fig. 1.

Our model comprises three major modules, namely, DBAN, sample selection, and learning model ensemble. The outputs of the model ensemble are taken as input for both the DBAN module and the sample selection module.

The first module, the DBAN, contains two sub-networks, namely, teacher network and student network. These two networks share most structure and parameters except the last softmax layer. The loss functions of the two networks are detailed in the following subsections.

The second module, sample selection, attempts to exclude training samples which are useless or even harmful for training. The sample selection is implemented via a selection gate. The selection gate is O_{test}, and its value is in the range of [0, 1]. Theoretically, the selection gate should be ideally added in the input of the entire learning system. In practice, the corresponding performance is better when O_{test} is added as a multiplier in the loss function.

The third module, ensemble learning, has two important roles in the entire learning framework. This module has two outputs (see Fig. 1), namely, O_{train}

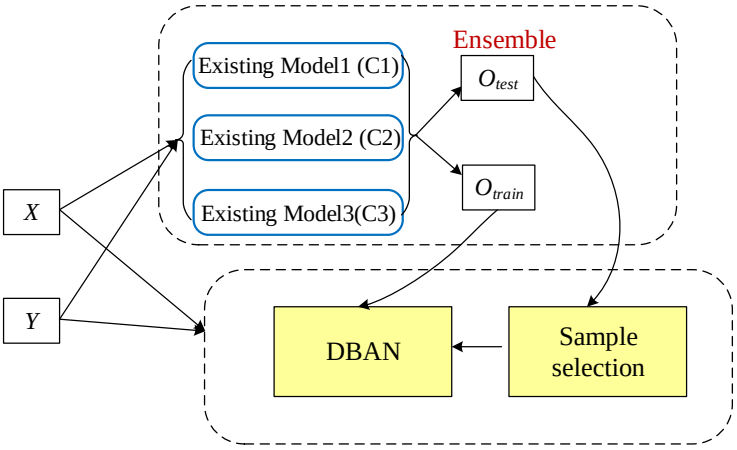


Fig. 1. Overview of the whole learning framework.

and O_{test}. The first output, O_{train}, is used as supplementary supervised information in DBAN, which is a standard operation in knowledge distillation. The second output, O_{test}, is used for sample selection.

3.2 Dually-born-again Network (DBAN)

The primary goal of knowledge distillation is to compress models while retaining the prediction performance as good as possible. BAN is a special type of knowledge distillation and its goal is to produce improved model parameters for an existing model rather than compressing the existing model. This study also aims to improve the performance of an existing sentiment analysis model without modifying the structure of the model.

The existing BAN merely involves one DNN structure to be learned. BAN iteratively borns a new student model. The newly born student model is subsequently used as the teacher model to born the next new student model. After K steps, a total of K student models are obtained. Experimental results on image classification indicate that the student network can achieve better results than the original model after several iterations. The graphical representation of the BAN training procedure is shown in Fig. 2. The kth model $f_{student}^{(k)}$ is trained based on the training samples X, true labels Y and the softmax output by the (k-1)th model $f_{student}^{(k-1)}$.

BAN needs to train the student network from scratch (all parameters are required to be re-initialized) in each step[1], leading to a high learning complexity. This work explores an alternatively training way in which teacher and student networks are trained simultaneously. Our dual learning procedure for network born again is called dually-born-again network (DBAN). Compared to the existing BAN, all parameters in DBAN are not required to be re-initialized and thus it has lower time consumption, which will be verified in the experimental section.

[1]Our experimental results indicate that the performance of the student network could not be improved if the parameters of the student network are initialized by copying those of its teacher network.

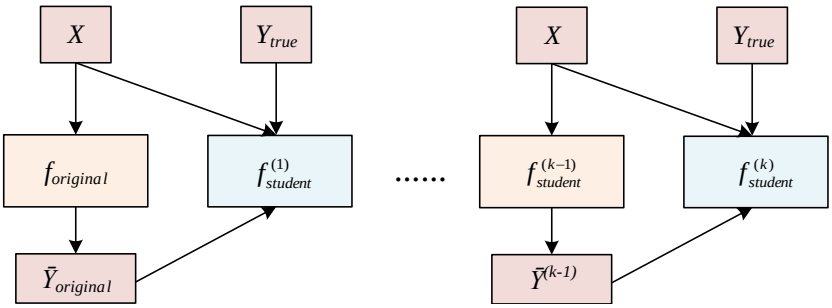


Fig. 2. The learning procedure of existing born-again network (BAN). The left image shows the first step and the right image shows the kth step.

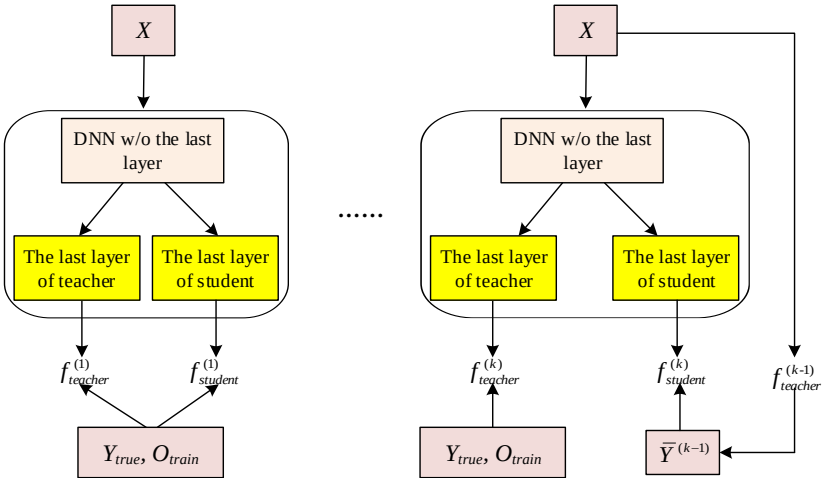


Fig. 3. The learning procedure of DBAN. The left shows the first step and the right shows the k-th step. Sample selection is considered in the loss function and does not explicitly appears in this figure.

The graphical model of DBAN is shown in Fig. 3. The teacher and the student networks share the same network structure and most of the parameters. Only the last layers of the two networks are different.

The final supervised labels for teacher network is the mixture of ground-truth labels Y_{true} and ensemble-based labels $Y_{ensemble}$.

In the initialization step, the teacher and student networks are trained based on the sample set X, ground-truth label set Y and O_{train} generated by the ensemble module. Thereafter, two models $f^{(1)}_{teacher}$ and $f^{(1)}_{student}$ are obtained. In this step, the loss function is defined as follows:

$$loss^{(1)} = \sum_i l(f^{(1)}_{teacher}(x_i), \widetilde{y}_i) + l(f^{(1)}_{student}(x_i), \widetilde{y}_i) \tag{1}$$

where

$$\widetilde{y}_i = (1 - \lambda_1)\ y_i + \lambda_1\ o_i^{(tr)} \tag{2}$$

where $y_i \in Y$ is the one-hot vector of the ground-truth label of the ith sample. $o_i^{(tr)} \in O_{train}$, and $\lambda_1 \in [0, 1]$ is a parameter.

Algorithm 1: Dually-born-again Network

Input : training set $\mathcal{D}$, number of training steps K, training configurations $\{\lambda_1, n\}$;

Initialize $\boldsymbol{\theta}_0$;
$k \leftarrow 1$;
Compute loss $loss^{(1)}$ using Eq. (1);
Updata $\boldsymbol{\theta}_0$ to $\boldsymbol{\theta}_1$ by training n iterations using $loss^{(1)}$;
Compute distribution output $f^{(2)}_{teacher}(x_i, \boldsymbol{\theta}_1)$;
for $k = 2, 3, \ldots, K$ **do**
    Compute loss $loss^{(k)}$ using Eq. (3);
    Updata $\boldsymbol{\theta}_{k-1}$ to $\boldsymbol{\theta}_k$ by training n iterations using $loss^{(k)}$;
    Compute distribution output $f^{(k)}_{teacher}(x_i, \boldsymbol{\theta}_k)$;
end

Return: $\mathbb{M} : \mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta} = \boldsymbol{\theta}_K)$.

In the kth step, the teacher network remains trained based on Y and O_{train}, whereas the student work is trained based on a new label set $Y^{(k-1)}$. The set $Y^{(k-1)}$ is the distribution output by running $f^{(k-1)}_{teacher}$ on X. The loss function used in the k-th step becomes the following form:

$$\begin{aligned} loss^{(k)} = & \textstyle\sum_i l(f^{(k)}_{teacher}(x_i), \widetilde{y}_i) \\ & + l(f^{(k)}_{student}(x_i), f^{(k-1)}_{teacher}(x_i)) \end{aligned} \tag{3}$$

In the implementation of the network training, the first step runs n iterations based on the loss defined in Eq. (1). Subsequently, the second step iteratively runs based on the loss defined in Eq. (3) until certain stop criteria is attained. The pseudo code of DBAN is provided in Algorithm 1.

In addition to the difference in parameter initialization strategy for student network training, there are two major differences between BAN and DBAN, as listed below:

- In DBAN, the teacher network and the student network are trained simultaneously. However, BAN trains the teacher network and the student network separately.
- In DBAN, the teacher network and the student network share most parameters. In BAN, the parameters between the teacher network and the student network are independent.

3.3 Sample Selection

Classification errors are inevitable in machine learning. The reason that classification errors of an involved model occurs mainly lies in the following aspects: (1) The classification capability of a model is not ideal. In practice, it is nearly impossible to construct a perfect model with 100% classification accuracy. (2) A few samples may play a negative role in training because several text semantics are reasonably vague or obscure to understand even by human beings. (3) A few labels are errors. Label inconsistency and labeling errors are inevitable in

Algorithm 2: Sample Selection

Input : training set $\mathcal{D}\{X, Y\}$;
Output: O_{test};
Evenly divided $\mathcal{D}$ into four subsets: $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$;
for $i = 1, 2, 3, 4$ **do**
    Train the model using ($\mathcal{D}$ - $\mathcal{D}_i$);
    Compute the maximum value of the softmax output on $\mathcal{D}_i$ $O_{test,i}$;
end
return $O_{test} = O_{test,1} \cup O_{test,2} \cup O_{test,3} \cup O_{test,4}$.

text sentiment annotation. These samples are harmful for the model learning. Intuitively, if text samples that are vague/obscure semantics or labeling errors are known in mode training, then these difficult-to-train samples can be excluded or given substantially low weights during the training stage. Therefore, the performance can be improved.

In order to identify the difficult-to-train samples, we made statistics on the error rates in terms of the maximum value of the softmax output on test data. The results shown in Fig. 4. The y-axis in Fig. 4 shows the error rates, while the x-axis shows the range of the maximum value of the softmax output. An increasing trend can be observed in Fig. 4. The error-classified samples are usually with lower maximum values of the corresponding softmax outputs. Alternatively, the maximum distribution outputs partially reflect the samples which are difficult to classify, thereby motivating us to conduct (difficult-to-train) sample selection based on maximum distribution output. The pseudo code of sample selection for one model is provided in Algorithm 2.

3.4 The Improved Loss with Sample Selection

By considering sample selection $o_i^{(te)} \in O_{test}$, the loss functions defined in Eqs. (1) and (3) can be improved as follows:

$$\begin{aligned} loss^{(1)} = &\textstyle\sum_i o_i^{(te)} \times l(f^{(1)}_{teacher}(x_i), \widetilde{y}_i) \\ &+ l(f^{(1)}_{student}(x_i), \widetilde{y}_i) + \lambda_2 |1 - o_i^{(te)}|_1 \end{aligned} \tag{4}$$

$$\begin{aligned} loss^{(k)} = &\textstyle\sum_i o_i^{(te)} \times l(f^{(k)}_{teacher}(x_i), \widetilde{y}_i) \\ &+ l(f^{(k)}_{student}(x_i), f^{(k-1)}_{teacher}(x_i)) + \lambda_2 |1 - o_i^{(te)}|_1 \end{aligned} \tag{5}$$

The last term is a sparse constraint that is used to prevent the loss from becoming zero if all the samples' selection-gate values equal to zero. The sparse constraint is reasonable because the proportion for the samples unsuitable for training is usually small.

3.5 Model Ensemble

Ensemble learning is effective toward improving prediction performances without applying novel models. It has been used in a number of NLP tasks [25]. Although an ensemble of models is usually more effective than a single model, it is usually in larger-size and more time-consuming. Therefore, this study does

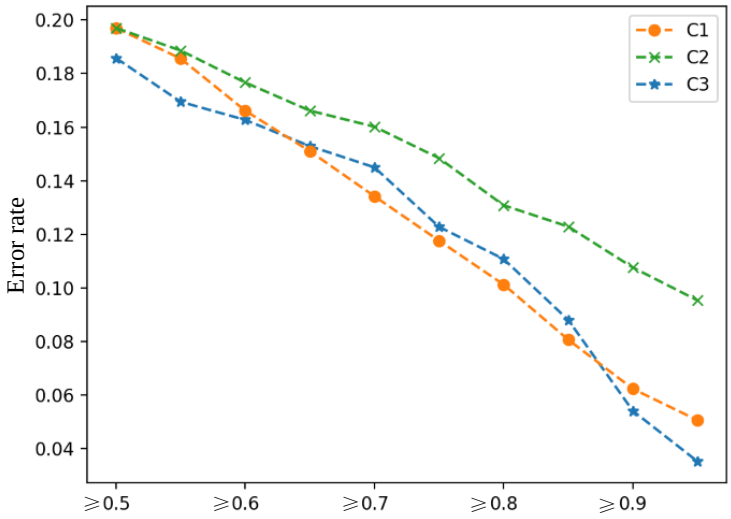


Fig. 4. The relationships between the maximum value of the softmax ouput and the errors.

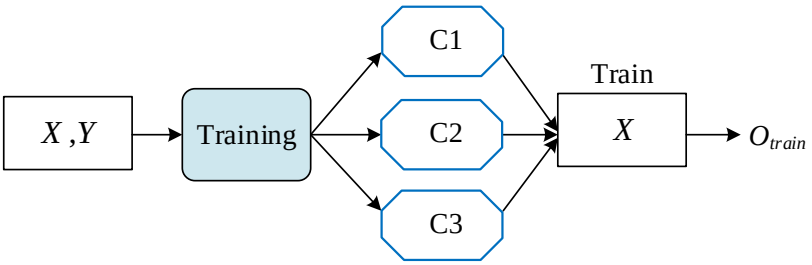
not attempt to design an ensemble for the final sentiment analysis. Instead, we use a single model for our final sentiment analysis while utilizing the ensemble in the training stage. As previously mentioned, ensemble learning plays two important roles in the whole learning framework according to its two different outputs. The first role is the supervised information for DBAN; the second role is for sample selection. These two roles are implemented by two forms of outputs of the ensemble models, i.e., O_{train} and O_{test} in Fig. 1.

Fig. 5 explains how outcomes O_{train} and O_{test} are generated based on the training data $\{X, Y\}$ and three existing sentiment analysis DNN models (C1, C2, and C3)[2]. As shown in Fig. 5(a), the data $\{X, Y\}$ are first used to train the three models, C1, C2, and C3. Then the three trained models are run on training sample set X to generate O_{train}. Let $o_{i1}^{(tr)}$, $o_{i2}^{(tr)}$, and $o_{i3}^{(tr)}$ be the three (distribution) labels output by the three trained models for $x_i \in X$. Then the generated label $o_i^{(tr)} \in O_{train}$ is as follow:

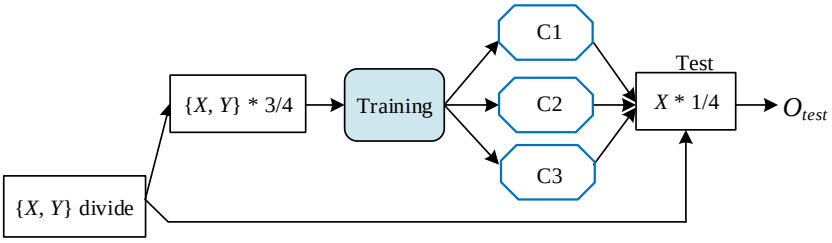
$$o_i^{(tr)} = (o_{i1}^{(tr)} + o_{i2}^{(tr)} + o_{i3}^{(tr)})/3. \tag{6}$$

In Fig. 5(b), the data $\{X, Y\}$ is first evenly divided into four subsets. Any combination of three subsets are then used to train three models, and the the rest subset is used to run the three models. After four iterations (note that the number of possible combination is four), O_{test} is generated. The selection gate $o_i^{(te)} \in O_{test}$ is constructed based on the output by the ensemble module. Let $o_{i,1}^{(te)}$, $o_{i,2}^{(te)}$, and $o_{i,3}^{(te)}$ be the three output distribution labels for the ith sample. Let $m(v)$ be the maximum value of the elements in a vector v. The selection

[2]The three models here is just for illustration. The number of models can also be 4, 5, etc..



(a) The generation of O_{train}.



(b) The generation of O_{test}.

Fig. 5. The approaches of generating O_{train} and O_{test}.

gate $o_i^{(te)}$ can be calculated as follows:

$$\begin{aligned} o_i^{(te)} &= w_1\, m(o_{i,1}^{(te)}) + w_2\, m(o_{i,2}^{(te)}) + w_3\, m(o_{i,3}^{(te)}) \\ s.t.\ & w_1 + w_2 + w_3 = 1;\ w_1, w_2, w_3 \geq 0 \end{aligned} \tag{7}$$

or

$$o_i^{(te)} = sigmoid(\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} m(o_{i,1}^{(te)}) \\ m(o_{i,2}^{(te)}) \\ m(o_{i,3}^{(te)}) \end{bmatrix} + b) \tag{8}$$

where w_1, w_2, w_3, and b are the parameters to be learned. The above two calculations are linear and non-linear, respectively.

4 Experiments

4.1 Data Sets

To demonstrate the effectiveness of our proposed method, as most previous works [3, 11, 26], we conduct experiments on two benchmarks, namely, movie reviews (MR) [27] and Stanford Sentiment Treebank (SST) [28]. Table 1 shows statistics of the two datasets.

MR: MR is a collection of movie reviews in English. Each sample in the MR dataset is divided into two categories, namely, negative and positive.

SST: The original SST data set provides phrase-level annotations. However, our experiments only considered the sentence-level annotations. Each sentence

Table 1: Details of the experimental data sets. Train/Dev/Test: train/development/test set.

DATA	MR	SST
Train	8636	8534
Dev	960	1100
Test	1066	2210

are classified into five categories (i.e. very positive, positive, neutral, negative, very negative).

4.2 Competing Models

In order to comprehensively evaluate the performance of our proposed method, the following state-of-the-art sentiment analysis DNN models are considered for comparison:

CNN [3]: This model utilizes convolution and pooling operations for sentiment analysis. In our experiment, the word vectors are static. Moreover, we added additional L2 penalty for the weights in last layer to reduce over fitting.

LSTM-CNN [22]:This model combines the CNN and LSTM networks to perform sentiment analysis. LSTM is used to encode the input layer, while CNN is used to obtain more higher-level representations on the hidden vectors.

LR-LSTM [11]: This model is a linguistical regularized variant of LSTM. Linguistically regularizer is based on lexical cues (e.g., polar words and negation words). This method achieves the state-of-the-art performances.

BiLSTM-ELMo-ATT [29]: ELMo (Embeddings from Language Models) generalizes traditional word embedding research along a different dimension. They propose to extract context sensitive features from a language model.

BERT$_{\mathbf{BASE}}$ [30]: BERT is based on a multi-layer bidirectional Transformer, and is trained on plain text for masked word prediction and next sentence prediction tasks. In order to apply a pre-trained model to specific natural language understanding tasks, we often need to fine-tune. BERT$_{\mathrm{BASE}}$ is the base BERT model released by the authors. Our implementation is based on the Tensorflow implementation of BERT [3].

4.3 Hyper-parameters and Training

All word vectors in our experiments are initialized by Glove [31]. The vocabulary size is 1.9M, while the dimension of each word vector is 300. The setting of hyper-parameters is mainly based on the original papers. All the LSTM hidden states are set to 300. The dropout rate is 0.5. The L2 regularization of CNN is 0.8, while that of LSTM-CNN is 0.4. The learning rate of CNN and LSTM-CNN is set to 0.01, while the batch size is set to 64. For LR-LSTM, the learning rate is set to 0.1, while the batch size is 25. The code of LR-LSTM is released by Qian et al. [11]. The other methods are implemented by Tensorflow[4]. The parameter λ_1 is searched in $[0, 0.2, 0.4, 0.6, 0.8, 1]$ and the parameter λ_2 is searched in $[0.0001, 0.001, 0.01, 0.1, 1, 10]$.

[3] https://github.com/google-research/bert#pre-trained-models

[4] https://www.tensorflow.org/

4.4 Overall Results

Table 2: The accuracies of the competing models on MR and SST.

Model	MR	SST
CNN	80.3	45.6
S+CNN	81.5	46.4
D+CNN	81.6	46.8
SD+CNN	**82.5**	**47.1**
LSTM-CNN	80.3	46.0
S+LSTM-CNN	82.3	47.4
D+LSTM-CNN	83.7	48.6
SD+LSTM-CNN	**84.4**	**49.3**
LR-LSTM	81.3	47.4
S+LR-LSTM	82.8	47.8
D+LR-LSTM	82.2	48.1
SD+LR-LSTM	**83.6**	**48.5**
BiLSTM-ELMo-ATT	81.6	48.4
S+BiLSTM-ELMo-ATT	83.0	49.1
D+BiLSTM-ELMo-ATT	82.1	48.8
SD+BiLSTM-ELMo-ATT	**83.4**	**49.7**
$\text{BERT}_{\text{BASE}}$	86.3	51.3
S+$\text{BERT}_{\text{BASE}}$	86.9	51.8
D+$\text{BERT}_{\text{BASE}}$	86.7	51.8
SD+$\text{BERT}_{\text{BASE}}$	**87.0**	**52.1**

Our proposed learning framework consists of several new modules (i.e., ensemble, DBAN, sample selection). By training existing DNN models using our proposed learning framework, the following new models are obtained[5]:

- **MODEL***: MODEL* indicates the baseline models without sample selection or DBAN.
- **S+MODEL***: This model is generated by training MODEL* with sample selection.
- **D+MODEL***: This model is generated by training MODEL* with DBAN.
- **SD+MODEL***: This model is generated by training MODEL* with the proposed learning framework (including both sample selection and DBAN).

Table 2 shows the experimental results of competing models on the two benchmark data sets MR and SST. The results verify that the classification accuracies of existing models (i.e., CNN, LSTM-CNN, LR-LSTM, ELMo and

[5]Note that the ensemble module is the basis module for both sample selection and DBAN. It is not independently used here, whereas its effectiveness is verified in an independent subsection.

$\text{BERT}_{\text{BASE}}$) are further improved by integrating the proposed learning framework. The model, SD+$\text{BERT}_{\text{BASE}}$, achieves the best results on both data sets. In addition, both the DBAN and sample selection are also demonstrated to be useful for the training of existing models.

On the MR data set, SD+CNN achieved an accuracy rate 2.2% points higher than the existing CNN model. Meanwhile, SD+LSTM-CNN model performed 4.1% better than the existing LSTM-CNN model, while the SD+LR-LSTM model performed 2.3% better than the existing LR-LSTM model. Even in the strong baseline models ELMo and $\text{BERT}_{\text{BASE}}$, our method still achieves 1.8% and 0.7% improvement respectively. On the SST data set, the accuracy rate is improved by 1.2, 2.6, 1.1, 1.3 and 0.8 percentage points compared with the existing CNN, LSTM-CNN, LR-LSTM, ELMo, $\text{BERT}_{\text{BASE}}$ models respectively.

Note that for selection sample, the results of using Eq. (7) are slightly better than those of using Eq. (8). Therefore, Eq. (7) is used in all experiments.

Table 3: The training time of DBAN and BAN.

Model	MR		SST	
	DBAN	BAN	DBAN	BAN
CNN	∼2.5h	∼4.8h	∼2.8h	∼5.0h
LSTM-CNN	∼4.1h	∼8.0h	∼4.5h	∼9.0h
LR-CNN	∼23h	∼40h	∼28h	∼45h
BiLSTM-ELMo-ATT	∼3.2h	∼6.1h	∼3.8h	∼7.1h
$\text{BERT}_{\text{BASE}}$	∼4.2h	∼7.6h	∼4.6h	∼8.7h

Table 4: The accuracy comparison between DBAN and BAN.

Model	MR	SST
CNN	80.3	45.6
BAN+CNN	81.4	46.7
D+CNN	**81.6**	**46.8**
LSTM-CNN	80.3	46.0
BAN+LSTM-CNN	82.8	48.4
D+LSTM-CNN	**83.7**	**48.6**
LR-LSTM	81.3	47.4
BAN+LR-LSTM	82.0	47.6
D+LR-LSTM	**82.2**	**48.1**
BiLSTM-ELMo-ATT	81.6	48.4
BAN+BiLSTM-ELMo-ATT	81.9	48.7
D+BiLSTM-ELMo-ATT	**82.1**	**48.8**
$\text{BERT}_{\text{BASE}}$	86.3	51.3
BAN+$\text{BERT}_{\text{BASE}}$	86.4	51.8
D+$\text{BERT}_{\text{BASE}}$	**86.7**	**51.8**

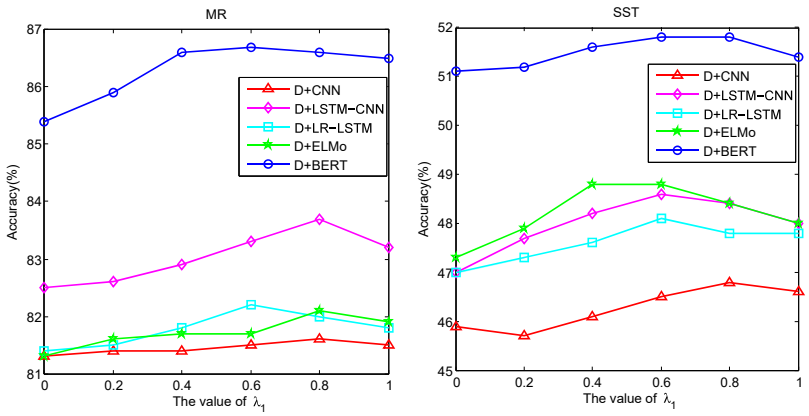


Fig. 6. The variations of the accuracies under different values of λ_1 in Eq. (3).

4.5 Comparison of DBAN and BAN

To further reveal the effectiveness of DBAN, we compared DBAN and BAN with the five existing DNN models, namely, CNN, LSTM-CNN, LR-LSTM, ELMo and $\text{BERT}_{\text{BASE}}$. The ensemble module (defined in Eq. (6)) is used in both DBAN and BAN.

The results shown in Table 3 indicate that DBAN is generally better than BAN. As previous stated, the training with BAN is time consumption as the network parameters should be re-initialized in each step. The training time of DBAN and BAN is recorded during learning and the results are shown in Table 4. Therefore, DBAN requires less training time than BAN, and the performance is better than BAN.

4.6 Experiments on the Ensemble Module

In DBAN, the output (O_{train}) of the ensemble module is used as the additional supervised information defined in Eq. (2). The parameter λ_1 in Eq. (2) tunes the proportion of the ground-truth labels and the labels O_{train}. Fig. 6 shows the performances of D+CNN, D+LSTM-CNN, D+LR-LSTM, D+ELMo and $\text{D+BERT}_{\text{BASE}}$ under different values of λ_1. The results show that with λ_1 increases, the accuracy increases as well. Nevertheless, when λ_1 is closed to 1, the accuracy decreases. The variations indicate that O_{train} plays more important role in DBAN than the ground-truth labels ($\lambda_1 = 0$). The ensemble output O_{train} benefits the performances of DBAN.

In order to verify the influence of ensemble learning in sample selection, we conducted experiments on the two benchmark datasets by using different inputs for the sample selection module. The results are shown in Table 5. When O_{test} is generated by ensemble module, the corresponding models achieve the highest accuracies for each model on both the data sets. The results also verify the usefulness of the ensemble module.

Table 5: The accuracy comparison when different inputs for sample selection are utilized.

Model	Input for sample selection	MR	SST
CNN	Without sample selection	80.3	45.6
	With sample selection	80.7	45.8
	Ensemble (O_{test})	**81.5**	**46.4**
LSTM-CNN	Without sample selection	80.3	46.0
	With sample selection	81.8	46.5
	Ensemble (O_{test})	**82.3**	**47.4**
LR-LSTM	Without sample selection	81.3	47.4
	With sample selectio	82.3	47.6
	Ensemble (O_{test})	**82.8**	**47.8**
BiLSTM-ELMo-ATT	Without sample selection	81.6	48.4
	With sample selectio	82.9	48.6
	Ensemble (O_{test})	**83.0**	**49.1**
$\text{BERT}_{\text{BASE}}$	Without sample selection	86.3	51.3
	With sample selectio	86.8	51.6
	Ensemble (O_{test})	**86.9**	**51.8**

4.7 Case Study for Sample Selection

Sample selection has rarely been analyzed in previous studies. To further illustrate that a few training samples play a negative role in training, we cite several typical examples from training sets, the weights of which are extremely minimal.

- "I've never bought from telemarketers, but I bought this movie". The weight value is 0.002. The review contains implicit information that indirectly expresses the affirmation of the movie.
- "The biggest problem with this movie is that its not nearly long enough". The weight value is 0.025. The text implicitly expresses the love for the movie, and does not actually say that the movie is considerably short.
- "The movie is not as terrible as the synergistic impulse that created it." The sentence contains negative words, and the weight value is 0.265. It implies that the model cannot considerably judge the sentiment of the text containing negative words.

The sentiment labels of the preceding samples are difficult to judge. That is, these samples add considerable burden to the training. Hence, reducing the weights of these samples benefits the training because "less is more" in real applications.

5 Conclusion

This paper investigates a new learning strategy to increase the sentiment analysis accuracy of a given deep neural network. Our proposed learning strategy is mainly based on BAN learning which is a special case of knowledge distillation. To mitigate the defects of existing BAN, a new learning approach, namely, DBAN, is presented and the teacher and student networks are trained simultaneously. The ensemble of existing DNN models is used for two goals. The

first goal is to provide substantially effective knowledge to network training in DBAN; the second goal is to perform sample selection to improve the quality of the training data. The experimental results verify the effectiveness of the proposed learning strategy as well as the three independent modules, namely, DBAN, sample selection and ensemble. The performances of three typical DNN models are enhanced after using our learning strategy.

In our future work, we will extend the proposed learning framework for more NLP tasks such as opinion mining and machine translation.

6 Acknowledgments

This work is supported by the Frontier science and technology innovation project (2019QY2404), Zhejiang Lab Fund2019KB0AB03), and Tianjin Nature Science Fund (19JCZDJC31300).

References

[1] S. R. Ahmad, A. A. Bakar, M. R. Yaakub, A review of feature selection techniques in sentiment analysis, Intell. Data Anal. 23 (1) (2019) 159–189.

[2] B. K. Norambuena, E. F. Lettura, C. M. Villegas, Sentiment analysis and opinion mining applied to scientific paper reviews, Intell. Data Anal. 23 (1) (2019) 191–214.

[3] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1746–1751.

[4] A. Severyn, A. Moschitti, Twitter sentiment analysis with deep convolutional neural networks, in: The International ACM SIGIR Conference, 2015, pp. 959–962.

[5] C. N. D. Santos, M. Gattit, Deep convolutional neural networks for sentiment analysis of short texts, in: International Conference on Computational Linguistics, 2014.

[6] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Meeting of the Association for Computational Linguistics, 2014, pp. 1555–1565.

[7] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: Conference on Empirical Methods in Natural Language Processing, 2017, pp. 606–615.

[8] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: International Conference on Computational Linguistics, 2015, pp. 3298–3307.

[9] T. Gui, Q. Zhang, L. Zhao, Y. Lin, M. Peng, J. Gong, X. Huang, Long short-term memory with dynamic skip connections, in: The Thirty-Third Conference on the Association for the Advance of Artificial Intelligence, 2019, pp. 6481–6488.

[10] Y. Choi, C. Cardie, Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification, in: Conference on Empirical Methods in Natural Language Processing, 2009, pp. 590–598.

[11] Q. Qian, M. Huang, J. Lei, X. Zhu, Linguistically regularized lstms for sentiment classification, in: In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1679–1689.

[12] C. Yang, L. Xie, S. Qiao, A. Yuille, Knowledge distillation in generations: More tolerant teachers educate better students, in: In The IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[13] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, A. Anandkumar, Born again neural networks, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 1602–1611.

[14] V. S. Sheng, F. Provost, P. G. Ipeirotis, Get another label? improving data quality and data mining using multiple, noisy labelers, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, Usa, August, 2008, pp. 614–622.

[15] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, L. Li, Learning from noisy labels with distillation, in: IEEE International Conference on Computer Vision, 2017, pp. 1928–1936.

[16] T. G. Dietterichl, Ensemble learning, Handbook of Brain Theory and Neural Networks (2002) 125–142.

[17] M. Freitag, Y. Alonaizan, B. Sankaran, Ensemble distillation for neural machine translation, arXiv preprint arXiv:1702.01802 (2017).

[18] C. Bucilu, R. Caruana, A. Niculescu-mizil, Model compression, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pp. 535–541.

[19] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, Computer Science 14 (7) (2015) 38–39.

[20] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, In Advances in neural information processing systems (2017) 1195–1204.

[21] Y. Kim, A. M. Rush, Sequence-level knowledge distillation, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2016, pp. 1317–1327.

[22] P. M. Sosa, Twitter sentiment analysis using combined lstm-cnn models (2017).

[23] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, Training convolutional networks with noisy labels, In The 3th International Conference on Learning Representations (2015).

[24] C. E. Brodley, M. A. Friedl, Identifying mislabeled training data, Journal of Artificial Intelligence Research 11 (1) (2011) 131–167.

[25] C. C. Aggarwal, C. X. Zhai, A survey of text classification algorithms, in: Mining Text Data, 2012, pp. 163–222.

[26] Y. Wang, A. Sun, J. Han, Y. Liu, X. Zhu, Sentiment analysis by capsules, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, 2018, pp. 1165–1174.

[27] P. Bo, L. Lee, Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales (2005) 115–124.

[28] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank (2013) 1631–1642.

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227–2237.

[30] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

[31] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.