Learning with Noisy Labels for Sentence-level Sentiment Classification

Hao Wang [♮,‡] **Bing Liu** [‡,*] **Chaozhuo Li** [§] **Yan Yang** [♮] **Tianrui Li** [♮]
[♮]School of Information Science and Technology, Southwest Jiaotong University
hwang@my.swjtu.edu.cn, {yyang;trli}@swjtu.edu.cn
[‡]Department of Computer Science, University of Illinois at Chicago
liub@uic.edu
[§]State Key Lab of Software Development Environment, Beihang University
lichaozhuo@buaa.edu.cn

Abstract

Deep neural networks (DNNs) can fit (or even over-fit) the training data very well. If a DNN model is trained using data with noisy labels and tested on data with clean labels, the model may perform poorly. This paper studies the problem of learning with noisy labels for sentence-level sentiment classification. We propose a novel DNN model called NETAB (as shorthand for convolutional neural **NET**works with **AB**-networks) to handle noisy labels during training. NETAB consists of two convolutional neural networks, one with a noise transition layer for dealing with the input noisy labels and the other for predicting 'clean' labels. We train the two networks using their respective loss functions in a mutual reinforcement manner. Experimental results demonstrate the effectiveness of the proposed model.

1 Introduction

It is well known that sentiment annotation or labeling is subjective (Liu, 2012). Annotators often have many disagreements. This is especially so for crowd-workers who are not well trained. That is why one always feels that there are many errors in an annotated dataset. In this paper, we study whether it is possible to build accurate sentiment classifiers even with noisy-labeled training data. Sentiment classification aims to classify a piece of text according to the polarity of the sentiment expressed in the text, e.g., *positive* or *negative* (Pang and Lee, 2008; Liu, 2012; Zhang et al., 2018). In this work, we focus on sentence-level sentiment classification (SSC) with labeling errors.

As we will see in the experiment section, noisy labels in the training data can be highly damaging, especially for DNNs because they easily fit the training data and memorize their labels even when training data are corrupted with noisy labels (Zhang et al., 2017). Collecting datasets annotated with clean labels is costly and time-consuming as DNN based models usually require a large number of training examples. Researchers and practitioners typically have to resort to crowdsourcing. However, as mentioned above, the crowdsourced annotations can be quite noisy.

Research on learning with noisy labels dates back to 1980s (Angluin and Laird, 1988). It is still vibrant today (Mnih and Hinton, 2012; Natarajan et al., 2013, 2018; Menon et al., 2015; Gao et al., 2016; Liu and Tao, 2016; Khetan et al., 2018; Zhan et al., 2019) as it is highly challenging. We will discuss the related work in the next section.

This paper studies the problem of learning with noisy labels for SSC. Formally, we study the following problem.

Problem Definition: Given noisy labeled training sentences $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, where $x_i|_{i=1}^n$ is the i-th sentence and $y_i \in \{1, ..., c\}$ is the sentiment label of this sentence, the noisy labeled sentences are used to train a DNN model for a SSC task. The trained model is then used to classify sentences with clean labels to one of the c sentiment labels.

In this paper, we propose a convolutional neural **NET**work with **AB**-networks (NETAB) to deal with noisy labels during training, as shown in Figure 1. We will introduce the details in the subsequent sections. Basically, NETAB consists of two convolutional neural networks (CNNs) (see Figure 1), one for learning sentiment scores to predict 'clean' [1] labels and the other for learning a noise transition matrix to handle input noisy labels. We call the two CNNs A-network and AB-network, respectively. The fundamental here is that (1) DNNs memorize easy instances first and gradu-

[*]Corresponding author

[1]Here we use clean with single quotes as it is not completely clean. In practice, models can hardly produce completely clean labels.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 6286–6292, Hong Kong, China, November 3–7, 2019.

ally adapt to hard instances as training epochs increase (Zhang et al., 2017; Arpit et al., 2017); and (2) noisy labels are theoretically flipped from the clean/true labels by a noise transition matrix (Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017; Han et al., 2018a,b). We motivate and propose a CNN model with a transition layer to estimate the noise transition matrix for the input noisy labels, while exploiting another CNN to predict 'clean' labels for the input training (and test) sentences. In training, we pre-train A-network in early epochs and then train AB-network and A-network with their own loss functions in an alternating manner. To our knowledge, this is the first work that addresses the noisy label problem in sentence-level sentiment analysis. Our experimental results show that the proposed model outperforms the state-of-the-art methods.

2 Related Work

Our work is related to sentence sentiment classification (SSC). SSC has been studied extensively (Hu and Liu, 2004; Pang and Lee, 2005; Zhao et al., 2008; Narayanan et al., 2009; Täckström and McDonald, 2011; Wang and Manning, 2012; Yang and Cardie, 2014; Kim, 2014; Tang et al., 2015; Wu et al., 2017; Wang et al., 2018). None of them can handle noisy labels. Since many social media datasets are noisy, researchers have tried to build robust models (Gamon, 2004; Barbosa and Feng, 2010; Liu et al., 2012). However, they treat noisy data as additional information and don't specifically handle noisy labels. A noise-aware classification model in (Zhan et al., 2019) trains using data annotated with multiple labels. Wang et al. (2016) exploited the connection of users and noisy labels of sentiments in social networks. Since the two works use multiple-labeled data or users' information (we only use single-labeled data, and we do not use any additional information), they have different settings than ours.

Our work is closely related to DNNs based approaches to learning with noisy labels. DNNs based approaches explored three main directions: (1) training DNNs on selected samples (Malach and Shalev-Shwartz, 2017; Jiang et al., 2018; Ren et al., 2018; Han et al., 2018b), (2) modifying the loss function of DNNs with regularization biases (Mnih and Hinton, 2012; Jindal et al., 2016; Patrini et al., 2017; Ghosh et al., 2017; Ma et al., 2018; Zhang and Sabuncu, 2018), and (3) plugging an extra layer into DNNs (Sukhbaatar et al., 2015; Bekker and Goldberger, 2016; Goldberger and Ben-Reuven, 2017; Han et al., 2018a). All these approaches were proposed for image classification where training images were corrupted with noisy labels. Some of them require noise rate to be known a priori in order to tune their models during training (Patrini et al., 2017; Han et al., 2018b). Our approach combines direction (1) and direction (3), and trains two networks jointly without knowing the noise rate. We have used five latest existing methods in our experiments for SSC. The experimental results show that they are inferior to our proposed method.

In addition, Xiao et al. (2015), Reed et al. (2015), Guan et al. (2016), Li et al. (2017), Veit et al. (2017), and Vahdat (2017) studied weakly-supervised DNNs or semi-supervised DNNs. But they still need some clean-labeled training data. We use no clean-labeled data.

3 Proposed Model

Our model builds on CNN (Kim, 2014). The key idea is to train two CNNs alternately, one for addressing the input noisy labels and the other for predicting 'clean' labels. The overall architecture of the proposed model is given in Figure 1. Before going further, we first introduce a proposition, a property, and an assumption below.

Proposition 1 *Noisy labels are flipped from clean labels by an unknown noise transition matrix.*

Proposition 1 is reformulated from (Han et al., 2018a) and has been investigated in (Sukhbaatar et al., 2015; Goldberger and Ben-Reuven, 2017; Bekker and Goldberger, 2016). This proposition shows that if we know the noise transition matrix, we can use it to recover the clean labels. In other words, we can put noise transition matrix on clean labels to deal with noisy labels. Given these, we ask the following question: *How to estimate such an unknown noise transition matrix*?

Below we give a solution to this question based on the following property of DNNs.

Property 1 *DNNs tend to prioritize memorization of simple instances first and then gradually memorize hard instances (Zhang et al., 2017).*

Arpit et al. (2017) further investigated this property of DNNs. Our setting is that simple instances are sentences of clean labels and hard instances are those with noisy labels. We also have the following assumption.

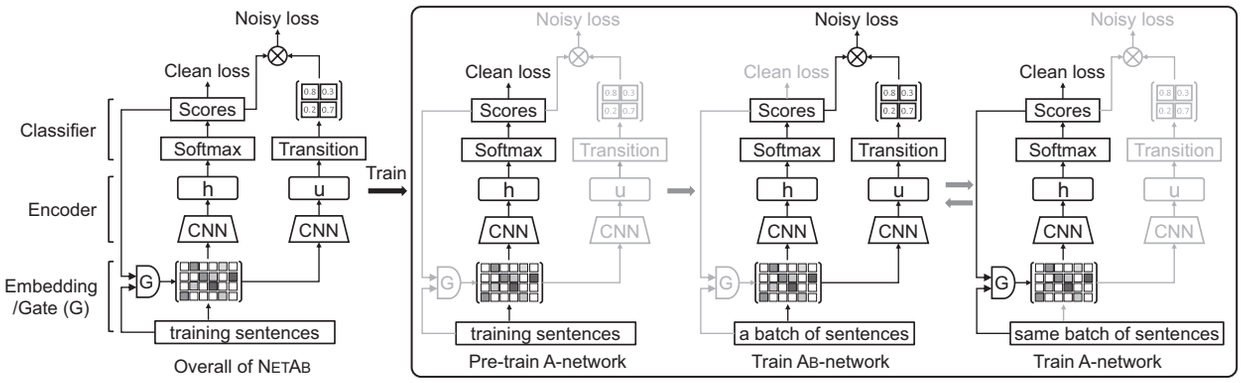

Figure 1: The proposed NETAB model (left) and its training method (right). Components in light gray color denote that these components are deactivated during training in that stage. (Color online)

Assumption 1 *The noise rate of the training data is less than* 50%.

This assumption is usually satisfied in practice because without it, it is hard to tackle the input noisy labels during training.

Based on the above preliminaries, we need to estimate the noisy transition matrix $Q \in \mathbb{R}^{c \times c}$ ($c = 2$ in our case, i.e., *positive* and *negative*), and train two classifiers $\ddot{y} \sim P(\ddot{y}|x, \theta)$ and $\widehat{y} \sim P(\widehat{y}|x, \vartheta)$, where x is an input sentence, $\ddot{y}$ is its noisy label, $\widehat{y}$ is its 'clean' label, θ and ϑ are the parameters of two classifiers. Note that both $\ddot{y}$ and $\widehat{y}$ here are the prediction results from our model, not the input labels. We propose to formulate the probability of the sentence x labeled as j with

$$P(\ddot{y}=j|x,\theta)=\sum_i P(\ddot{y}=j|\widehat{y}=i)P(\widehat{y}=i|x,\vartheta) \tag{1}$$

where $P(\ddot{y} = j|\widehat{y} = i)$ is an item (the ji-th item) in the noisy transition matrix Q. We can see that the noisy transition matrix Q is exploited on the 'clean' scores $P(\widehat{y}|x, \vartheta)$ to tackle noisy labels.

We now present our model NETAB and introduce how NETAB performs Eq. (1). As shown in Figure 1, NETAB consists of two CNNs. The intuition here is that we use one CNN to perform $P(\widehat{y} = i|x, \vartheta)$ and use another CNN to perform $P(\ddot{y} = j|x, \theta)$. Meanwhile, the CNN performing $P(\ddot{y} = j|x, \theta)$ estimates the noise transition matrix Q to deal with noisy labels. Thus we add a transition layer into this CNN.

More precisely, in Figure 1, the CNN with a clean loss performs $P(\widehat{y} = i|x, \vartheta)$. We call this CNN the A-network. The other CNN with a noisy loss performs $P(\ddot{y} = j|x, \theta)$. We call this CNN the AB-network. AB-network shares all the parameters of A-network except the parameters from the Gate unit and the clean loss. In addition, AB-network has a transition layer to estimate the noisy transition matrix Q. In such a way, A-network predict 'clean' labels, and AB-network handles the input noisy labels.

We use cross-entropy with the predicted labels $\ddot{y}$ and the input labels y (given in the dataset) to compute the noisy loss, formulated as below

$$\mathcal{L}_{noisy}=-\frac{1}{|\ddot{S}|}\sum_{x\in\ddot{S}}\sum_i \mathbb{I}(y=i|x)\log P(\ddot{y}=i|x) \tag{2}$$

where $\mathbb{I}$ is the indicator function (if $y == i$, $\mathbb{I} = 1$; otherwise, $\mathbb{I} = 0$), and $|\ddot{S}|$ is the number of sentences to train AB-network in each batch.

Similarly, we use cross-entropy with the predicted labels $\widehat{y}$ and the input labels y to compute the clean loss, formulated as

$$\mathcal{L}_{clean}=-\frac{1}{|\widehat{S}|}\sum_{x\in\widehat{S}}\sum_i \mathbb{I}(y=i|x)\log P(\widehat{y}=i|x) \tag{3}$$

where $|\widehat{S}|$ is the number of sentences to train A-network in each batch.

Next we introduce how our model learns the parameters (ϑ, θ and Q). An embedding matrix v is produced for each sentence x by looking up a pre-trained word embedding database (e.g., GloVe.840B (Pennington et al., 2014)). Then an encoding vector $h = CNN(v)$ (and $u = CNN(v)$) is produced for each embedding matrix v in A-network (and AB-network). A sofmax classifier gives us $P(\hat{y} = i|x, \vartheta)$ (i.e., 'clean' sentiment scores) on the learned encoding vector h. As the noise transition matrix Q indicates the transition values from clean labels to noisy labels, we com-

	#Noisy Training Data	#Clean Training Data	#Validation Data	#Test Data
Movie	13539P, 13350N	4265P, 4265N	105P, 106N	960P, 957N
Laptop	9702P, 7876N	1064P, 490N	33P, 20N	298P, 175N
Restaurant	8094P, 10299N	1087P, 823N	39P, 14N	339P, 116N

Table 1: Summary statistics of the datasets. Number of positive (P) and negative (N) sentences in (noisy and clean) training data, validation data, and test data. The second column shows the statistics of sentences extracted from the 2,000 reviews of each dataset. The last three columns show the statistics of the sentences in three clean-labeled datasets, see "Clean-labeled Datasets".

pute Q as follows

$$Q = [q_1; q_2] \quad (4)$$

$$q_i = softmax(g_i f_i),\ i = 1, 2 \quad (5)$$

$$g_i = tanh(W_i u + b_i) \quad (6)$$

where W_i is a trainable parameter matrix, b_i and f_i are two trainable parameter vectors. They are trained in the AB-network. Finally, $P(\ddot{y} = j|x, \theta)$ is computed by Eq. (1).

In training, NETAB is trained end-to-end. Based on Proposition 1 and Property 1, we pre-train A-network in early epochs (e.g., 5 epochs). Then we train AB-network and A-network in an alternating manner. The two networks are trained using their respective cross-entropy loss. Given a batch of sentences, we first train AB-network. Then we use the scores predicted from A-network to select some possibly clean sentences from this batch and train A-network on the selected sentences. Specifically speaking, we use the predicted scores to compute sentiment labels by $\arg\max_i\{\ddot{y} = i|\ddot{y} \sim P(\ddot{y}|x, \theta)\}$. Then we select the sentences whose resulting sentiment label equals to the input label. The selection process is marked by a Gate unit in Figure 1. When testing a sentence, we use A-network to produce the final classification result.

4 Experiments

In this section, we evaluate the performance of the proposed NETAB model. we conduct two types of experiments. (1) We corrupt clean-labeled datasets to produce noisy-labeled datasets to show the impact of noises on sentiment classification accuracy. (2) We collect some real noisy data and use them to train models to evaluate the performance of NETAB.

Clean-labeled Datasets. We use three clean labeled datasets. The first one is the movie sentence polarity dataset from (Pang and Lee, 2005). The other two datasets are laptop and restaurant datasets collected from SemEval-2016 [2]. The former consists of laptop review sentences and the latter consists of restaurant review sentences. The original datasets (i.e., Laptop and Restaurant) were annotated with aspect polarity in each sentence. We used all sentences with only one polarity (*positive* or *negative*) for their aspects. That is, we only used sentences with aspects having the same sentiment label in each sentence. Thus, the sentiment of each aspect gives the ground-truth as the sentiments of all aspects are the same.

For each clean-labeled dataset, the sentences are randomly partitioned into training set and test set with 80% and 20%, respectively. Following (Kim, 2014), We also randomly select 10% of the test data for validation to check the model during training. Summary statistics of the training, validation, and test data are shown in Table 1.

Noisy-labeled Training Datasets. For the above three domains (movie, laptop, and restaurant), we collected 2,000 reviews for each domain from the same review source. We extracted sentences from each review and assigned review's label to its sentences. Like previous work, we treat 4 or 5 stars as positive and 1 or 2 stars as negative. The data is noisy because a positive (negative) review can contain negative (positive) sentences, and there are also neutral sentences. This gives us three noisy-labeled training datasets. We still use the same test sets as those for the clean-labeled datasets. Summary statistics of all the datasets are shown in Table 1.

Experiment 1: Here we use the clean-labeled data (i.e., the last three columns in Table 1). We corrupt the clean training data by switching the labels of some random instances based on a noise rate parameter. Then we use the corrupted data to train NETAB and CNN (Kim, 2014).

The test accuracy curves with the noise rates [0, 0.1, 0.2, 0.3, 0.4, 0.5] are shown in Figure 2.

[2] http://alt.qcri.org/semeval2016/task5/

Methods	Movie			Laptop			Restaurant		
	ACC	F1_pos	F1_neg	ACC	F1_pos	F1_neg	ACC	F1_pos	F1_neg
NBSVM-uni (Wang and Manning, 2012)	0.6791	0.6663	0.6910	0.7637	0.8216	0.6500	0.7949	0.8478	0.6858
NBSVM-bi (Wang and Manning, 2012)	0.6416	0.6438	0.6394	0.7784	0.8320	**0.6749**	0.7154	0.7834	0.5853
CNN (Kim, 2014)	0.6667	0.6467	0.6844	0.7737	0.8381	0.6245	0.8329	0.8841	0.7007
Adaptation (Goldberger and Ben-Reuven, 2017)	0.6682	0.6708	0.6656	0.7272	0.7936	0.5981	0.8285	0.8872	0.6422
Forward (Patrini et al., 2017)	0.6864	0.6753	0.6969	0.7547	0.8170	0.6282	0.8329	0.8882	0.6695
Backward (Patrini et al., 2017)	0.6651	0.6160	0.6830	0.7124	0.7834	0.5723	0.7890	0.8485	0.6521
Masking (Han et al., 2018a)	0.6708	0.6631	0.6782	0.7188	0.7787	0.6144	0.8219	0.8789	0.6639
Co-teaching (Han et al., 2018b)	0.6150	0.5980	0.6306	0.7145	0.7867	0.5686	0.7978	0.8575	0.6515
NETAB (Our method)	**0.7047**	**0.7076**	**0.7017**	**0.7928**	**0.8487**	0.6711	**0.8593**	**0.9056**	**0.7241**

Table 2: Accuracy (ACC) of both classes, F1 (F1_pos) of positive class and F1 (F1_neg) of negative class on clean test data/sentences. Training data are real noisy-labeled sentences.

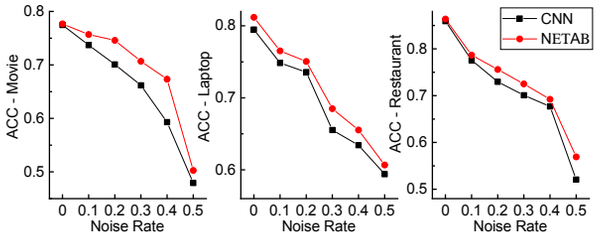

Figure 2: Accuracy (ACC) on clean test data. For training, the labels of clean data are flipped with the noise rates [0, 0.1, 0.2, 0.3, 0.4, 0.5]. For example, 0.1 means that 10% of the labels are flipped. (Color online)

From the figure, we can see that the test accuracy drops from around 0.8 to 0.5 when the noise rate increases from 0 to 0.5, but our NETAB outperforms CNN. The results clearly show that the performance of the CNN drops quite a lot with the noise rate increasing.

Experiment 2: Here we use the real noisy-labeled training data to train our model and the baselines, and then test on the test data in Table 1. Our goal is two fold. First, we want to evaluate NETAB using real noisy data. Second, we want to see whether sentences with review level labels can be used to build effective SSC models.

Baselines. We use one strong non-DNN baseline, NBSVM (with unigrams or bigrams features) (Wang and Manning, 2012) and six DNN baselines. The first DNN baseline is CNN (Kim, 2014), which does not handle noisy labels. The other five were designed to handle noisy labels.

The comparison results are shown in Table 2. From the results, we can make the following observations. (1) Our NETAB model achieves the best ACC and F1 on all datasets except for F1 of negative class on Laptop. The results demonstrate the superiority of NETAB. (2) NETAB outperforms the baselines designed for learning with noisy labels. These baselines are inferior to ours as they were tailored for image classification. Note that we found no existing method to deal with noisy labels for SSC.

Training Details. We use the publicly available pre-trained embedding GloVe.840B (Pennington et al., 2014) to initialize the word vectors and the embedding dimension is 300.

For each baseline, we obtain the system from its author and use its default parameters. As the DNN baselines (except CNN) were proposed for image classification, we change the input channels from 3 to 1. For our NETAB, we follow Kim (2014) to use window sizes of 3, 4 and 5 words with 100 feature maps per window size, resulting in 300-dimensional encoding vectors. The input length of sentence is set to 40. The network parameters are updated using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.001. The learning rate is clipped gradually using a norm of 0.96 in performing the Adam optimization. The dropout rate is 0.5 in the input layer. The number of epochs is 200 and batch size is 50.

5 Conclusions

This paper proposed a novel CNN based model for sentence-level sentiment classification learning for data with noisy labels. The proposed model learns to handle noisy labels during training by training two networks alternately. The learned noisy transition matrices are used to tackle noisy labels. Experimental results showed that the proposed model outperforms a wide range of baselines markedly. We believe that learning with noisy labels is a promising direction as it is often easy to collect noisy-labeled training data.

Acknowledgments

Hao Wang and Yan Yang's work was partially supported by a grant from the National Natural Science Foundation of China (No. 61572407).

References

Dana Angluin and Philip Laird. 1988. Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Devansh Arpit, Stanislaw K. Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron C. Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *ICML*, pages 233–242.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *COLING*, pages 36–44.

Alan Joseph Bekker and Jacob Goldberger. 2016. Training deep neural-networks based on unreliable labels. In *ICASSP*, pages 2682–2686.

Michael Gamon. 2004. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In *COLING*, pages 841–847.

Wei Gao, Lu Wang, Yu-Feng Li, and Zhi-Hua Zhou. 2016. Risk minimization in the presence of label noise. In *AAAI*, pages 1575–1581.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *AAAI*, pages 1919–1925.

Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *ICLR*, pages 1–9.

Ziyu Guan, Long Chen, Wei Zhao, Yi Zheng, Shulong Tan, and Deng Cai. 2016. Weakly-supervised deep learning for customer review sentiment classification. In *IJCAI*, pages 3719–3725.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Tvor Tsang, Ya Zhang, and Masashi Sugiyama. 2018a. Masking: A new perspective of noisy supervision. In *NIPS*, pages 5836–5846.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018b. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NIPS*, pages 8527–8537.

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

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. 2016. Learning deep networks from noisy labels with dropout regularization. In *ICDM*, pages 967–972.

Ashish Khetan, Zachary C Lipton, and Animashree Anandkumar. 2018. Learning from noisy singly-labeled data. In *ICLR*, pages 1–15.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1476–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *CVPR*, pages 1910–1918.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, pages 1678–1684.

Tongliang Liu and Dacheng Tao. 2016. Classification with noisy labels by importance reweighting. *TPAMI*, 38(3):447–461.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In *ICML*, pages 3361–3370.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update". In *NIPS*, pages 960–970.

Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. 2015. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pages 125–134.

Volodymyr Mnih and Geoffrey E Hinton. 2012. Learning to label aerial images from noisy data. In *ICML*, pages 567–574.

Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *EMNLP*, pages 180–189.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep Ravikumar, and Ambuj Tewari. 2018. Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research*, 18:1–33.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *NIPS*, pages 1196–1204.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *ICLR Workshop Track*, pages 1–11.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *ICML*, pages 4331–4340.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2015. Training convolutional networks with noisy labels. In *ICLR Workshop Track*, pages 1–11.

Oscar Täckström and Ryan McDonald. 2011. Semi-supervised latent variable models for sentence-level sentiment analysis. In *ACL*, pages 569–574.

Duyu Tang, Bing Qin, Furu Wei, Li Dong, Ting Liu, and Ming Zhou. 2015. A joint segmentation and classification framework for sentence level sentiment classification. *TASLP*, 23(11):1750–1761.

Arash Vahdat. 2017. Toward robustness against label noise in training deep discriminative neural networks. In *NIPS*, pages 5596–5605.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 839–847.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94.

Yaowei Wang, Yanghui Rao, Xueying Zhan, Huijun Chen, Maoquan Luo, and Jian Yin. 2016. Sentiment and emotion classification over noisy labels. *Knowledge-Based Systems*, 111:207–216.

Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment analysis by capsules. In *WWW*, pages 1165–1174.

Fangzhao Wu, Jia Zhang, Zhigang Yuan, Sixing Wu, Yongfeng Huang, and Jun Yan. 2017. Sentence-level sentiment classification with weak supervision. In *SIGIR*, pages 973–976.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699.

Bishan Yang and Claire Cardie. 2014. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *ACL*, pages 325–335.

Xueying Zhan, Yaowei Wang, Yanghui Rao, and Qing Li. 2019. Learning from multi-annotator data: A noise-aware classification framework. *ACM Trans. Inf. Syst.*, 37(2):26:1–26:28.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*, pages 1–15.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NIPS*, pages 8778–8788.

Jun Zhao, Kang Liu, and Gen Wang. 2008. Adding redundant features for CRFs-based sentence sentiment classification. In *EMNLP*, pages 117–126.