Sentiment analysis using semi-supervised learning with few labeled data

Yuhao Pan[1,2], Zhiqun Chen[1], Yoshimi Suzuki[2], Fumiyo Fukumoto[2], Hiromitsu Nishizaki[2]

[1]School of Computer Science and Technology,
Hangzhou Dianzi University,
HangZhou, China
{pyh, chenzq}@hdu.edu.cn

[2]Integrated Graduate School of Medicine, Engineering,
and Agricultural Sciences, Faculty of Engineering
University of Yamanashi
Kofu, Japan
{ysuzuki, fukumoto, hnishi}@yamanashi.ac.jp

***Abstract*—Sentiment analysis has been widely explored in many text domains, including tweets, movie reviews, shop/restaurant reviews, product reviews, and peer reviews for scholarly papers. However, it is very costly to manually label the training data for sentiment analysis. We focus on the problem and presents an approach for leveraging contextual features from unlabeled movie and restaurant reviews with a neural-network-based learning model, Ladder network. The experi-mental results by using two benchmark datasets, IMDb and YelpNYC, show that our model outperforms the baseline models including LSTM and SVM. Especially we verified that our model is better performance gaining on limited training datasets with 1% data labeled. Our source codes are available online. [1]**

Keywords-Ladder network; reviews; unlabeled; sentiment analysis

I. INTRODUCTION

Sentiment analysis is the classification task by sentiments, moods, attitudes, and subjectivities in a textual context. As an area of natural language processing (NLP), sentiment analysis leads a new perspective to treat traditional text classification. Nowadays, it is one of the most challenging research areas in NLP and is also widely studied in text mining. More recently, sentiment analysis based on deep learning techniques has been intensively studied. These attempts include bidirectional long short-term memory (LSTM), Convolutional Neural Network (CNN), and memory network. It enables to use various contexts which are powerful for learning features from the training data. However, deep learning techniques requires large amounts of labeled training data, which is not always available.

We focus on the problem of few labeled training data and presents an approach for leveraging contextual features from unlabeled data with a neural-network-based learning model, Ladder network.

Our learning framework consists of two models, i.e. (1) a model that pre-training text jointly conditioning on both left and the right context, word embedding. (2) an encoder-decoder model that unsupervised learn the sentiment information in reviews, Ladder network(LN) [1].

Ladder network is a solution of semi-supervised learning. Previous work demonstrated the interpretability of this encoder-decoder model. We compared our model with other several word embedding models and machine learning models on the IMDb [2] and YelpNYC [3] datasets.

The main contribution of our work can be summarized:

(1) We introduce Ladder networks which integrate a small amount of labeled data with a large number of unlabeled reviews and augment data effectively.
(2) We test our hypothesis that Ladder network helps to improve the overall performance of the sentiment analysis task with a very small amount of labeled training data.
(3) We optimize the Ladder network which used in image processing classification to sentiment analysis.
(4) We compare several BERT variants for sentiment analysis on two different datasets: IMDb and YelpNYC.

II. RELATED WORK

Sentiment analysis is one of the major topics of NLP. It is beneficial for many NLP applications such as marketing analysis and fake news detection [4]. An abstraction of the sentiment analysis is defined in Liu, Bing [5]. Studies of Liu show the details of sentiment analysis. SentiWordNet is a lexical resource for sentiment analysis (Esuli, A., & Sebastiani, F., 2006) [6]. Khan (Khan, et al., 2017) [7] led Information Gain and Cosine Similarity into SentiWordNet. It applies lexicon-based methodology with machine learning in the semi-supervised problem. However, their method only focuses on the lexical resource.

Previous studies on semi-supervised learning (Vincent P et al.,2008) [8] proposed an approach training denoising autoencoders. By stacking these autoencoder models can be motivated.

The model we present draws inspiration from prior work on semi-supervised learning method Ladder network (Rasmus, Antti, et al., 2015). This paper offers an insight of training to simultaneously minimize the combination of supervised and unsupervised cost functions by backpropagation. Ladder networks have successfully applied to image processing classification MNIST and CIFAR-10 with high performance. Pezeshki M et al. [9] explained that Ladder network uses the lateral connections and the application of noise which produce a powerful learning model. Andrew [10] used LSTM recurrent network in text classification. The basic idea is that the model learns parameters from unsupervised learning can be used as a starting point in supervised learning. But this paper ignored the number of labeled data that will influence the result of the model. Nagesh (2018) [11] focuses on the task of named entity classification (CoNLL-2003 shared task). With context, Ladder network can define the correct label and give a better

[1] Our source code can be obtained from https://github.com/jepyh/sentiment_analysis_few_labeled

2642-3596/20/$31.00 ©2020 IEEE
DOI 10.1109/CW49994.2020.00044

result compared with Explicit Pattern-based Bootstrapping and Label Propagation.

The audio event classification model (Dubey H, 2019) [12] is based on CNN embedding. It provides an idea of decision-making after feature extraction rather than only applying Ladder network.

III. MODEL

A. Word Embedding Layer

Learning contextual representations is one of the core techniques in sentiment analysis. One attempt is pre-trained contextualized language representations. Many researchers have attempted to learn contextualized language representations by pre-training a language model with a large amount of unannotated data. Such attempts include one–hot, Word2Vec, BERT, DistilBERT, and ALBERT. In our work, we use these word embedding methods to extract contextual features. We set it to first layer after input.

B. Encoders of Ladder Network

There are two encoders in Ladder network. One is clean, and another with noise. Each encoder network contains many units of the encoder. We found a small number of units would perform better. Following the standard unit of the encoder. Equation (1,2,3) illustrate the final sentiment class of review linear activation is given by the softmax, and in other layers, ReLU is used. The hidden layer is a mapping by two parameters $\alpha^{(l)}$ and $\beta^{(l)}$. The value $\tilde{z}^{(l)}$ here comes from Gaussian noise and normalization.

$$\phi(\cdot) = \begin{cases} softmax(\cdot), l = L \\ ReLU(\cdot),\ other \end{cases} \quad (1)$$

$$\tilde{h}^{(l)} = \phi\left(\alpha^{(l)}\left(\tilde{z}^{(l)} + \beta^{(l)}\right)\right) \quad (2)$$

$$\tilde{z}^{(l)} = N_B\left(W^{(l)}\tilde{h}^{(l-1)}\right) + n^{(l)} \quad (3)$$

Ladder network provides $\tilde{z}^{(l)}$to contain semantic features rather than Denoising Autoencoders [13] based on input x. Figure 1 illustrates the noise encoder and the clean encoder.

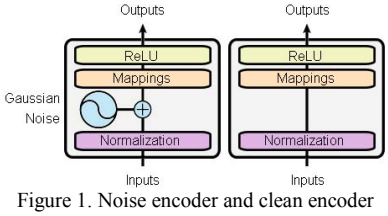

Figure 1. Noise encoder and clean encoder

C. Decoder of Ladder Network

Whether the clean encoder and noise encoder are stringers of Ladder, the decoder would be rail. $u_i^{(l)}$ is a projection vector, $V^{(l)}$ is the transpose of $W^{(l)}$ in (1). By calculating normalization we can get $u_i^{(l)}$ features from the previous layers. For denoising, we consider two functions in equation (4,5,6,7) where $a_i^{(l)}$ means trainable parameters. By $\tilde{z}_i^{(l)}$, $\mu_i(u_i^{(l)})$, $\upsilon_i(u_i^{(l)})$, model rebuild the value $\hat{z}_i^{(l)}$. $\varsigma(\cdot)$ means sigmoid function.

$$u^{(l)} = \begin{cases} \tilde{h}^{(l)}, & l = L \\ N_B\left(V^{(l+1)}\hat{z}^{(l+1)}\right), other \end{cases} \quad (4)$$

$$\mu_i(u_i^{(l)}) = a_{1,i}^{(l)}\varsigma\left(a_{2,i}^{(l)}u_i^{(l)} + a_{3,i}^{(l)}\right) + a_{4,i}^{(l)}u_i^{(l)} + a_{5,i}^{(l)} \quad (5)$$

$$\upsilon_i(u_i^{(l)}) = a_{6,i}^{(l)}\varsigma\left(a_{7,i}^{(l)}u_i^{(l)} + a_{8,i}^{(l)}\right) + a_{9,i}^{(l)}u_i^{(l)} + a_{10,i}^{(l)} \quad (6)$$

$$\hat{z}_i^{(l)} = \left(\tilde{z}_i^{(l)} - \mu_i(u_i^{(l)})\right)\upsilon_i(u_i^{(l)}) + \mu_i(u_i^{(l)}) \quad (7)$$

In Figure 2, at the bottom of the decoder, $\hat{x}$ is allowed to represent the meaning of input. Because of the difference between $\tilde{z}^{(l)}$ and $\hat{z}_i^{(l)}$, we can use LN to deal with little data being labeled.

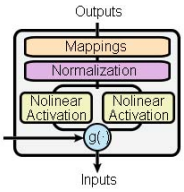

Figure 2. Decoder

D. Cost Function

The cost function is defined to minimize the difference between the clean encoder and noise encoder-decoder. By backpropagating, the cost is optimized to fit the model deal well with the sentiment classification.

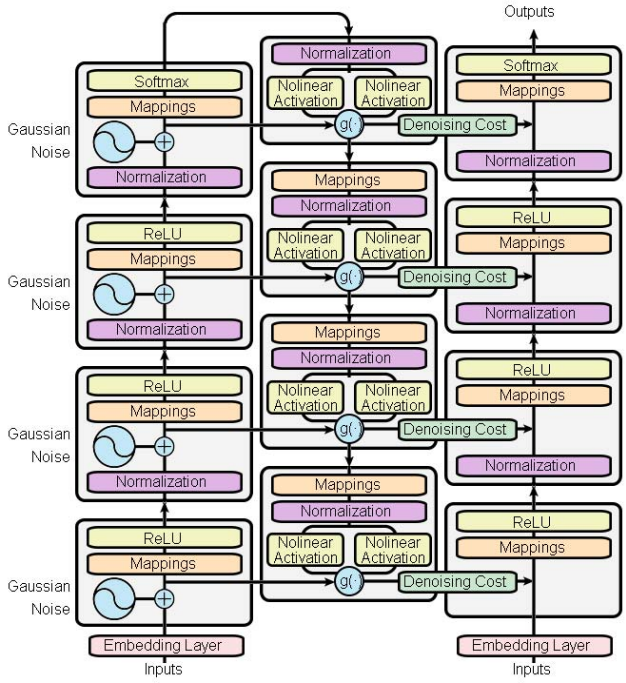

Figure 3. 4-layer Ladder network

Figure 3 illustrates the framework of our model. The blue boxes show the denoising cost function C_d. λ_l refers to a reconstruction cost weight of denoising cost. N indicates the sample number. m_l is each layer's width. $\hat{z}_{BN}^{(l)}$ stands for the normalization of $\hat{z}_i^{(l)}$(Equation 8).

$$C_d = \sum_{l=0}^{L}\frac{\lambda_l}{Nm_l}\sum_{n=0}^{N}(z^{(l)}(n) - \hat{z}_{BN}^{(l)}(n)) \quad (8)$$

The final total cost $C = C_c + C_d$. C_c is an average negative log probability of noising encoder's end to end.

IV. EXPERIMENT

A. Dataset

For experiments, we used two kinds of sentiment analysis datasets: IMDb and YelpNYC. IMDb is a large movie review dataset. It is a dataset for binary sentiment classification with positive and negative labels. IMDb dataset provides a set of 25,000 movie reviews for training and 25,000 for testing. The average length of reviews is 101 words.YelpNYC dataset includes 359,052 reviews for restaurants located in New York City. It provides a label of sentiment with a 1-5 stars level, as a 5-class classification problem. The average number of words in the reviews is 239.

B. Setting

For the variety length of reviews, we pick a 450 words window to collect input text for IMDb and 300 for YelpNYC. This is because 90% of IMDb dataset consists of reviews in less than 450 words, and 95.8% YelpNYC reviews in less than 300 words. Preprocessing including filtered out stopwords and non-English reviews as the number of 100,000 for training and 100,000 for testing. At the step of word embedding, we use the following models. One-hot model considers the topmost 5,000 high frequency words. Word2Vec [14] is based on Word2vec-GoogleNews-vectors with a dimension of 300. In BERT [15], we use two models: BERT-base-cased, BERT-large-cased. Besides, we do the experiments with the recent model DistilBERT-base-cased [16] and ALBERT [17] with no dropout. The details of these models are show in TABLE I.

TABLE I Details of the word embedding models

Embedding model	*Layer*	*Hidden*	*Head*
BERT-base	12	768	12
BERT-large	24	1024	16
DistilBERT	6	768	12
ALBERT	12	768	12

We used a 4-layer Ladder network with dimensions [INPUT_SIZE DIM_WE WIN_RV N], DIM_WE is the dimension of word embedding WIN _RV is the window to collect text. N is the number of classes on the dataset. We set Gaussian noise to std = 0.3 and denoising cost weight to [1000, 10, 0.1, 0.1]. Our experiment shows that a small number of layers play a crucial role in LN rather stacking of a large number of network layers.

C. Main results

With the preparing above we did a series of experiments. Labeled data is considered 0.5% to 4% as the few labeled in real-world datasets. Furthermore, we list one column with fully-labeled data as comparison on supervised learning. The results show even if using 4%, we can get the result comparable to 100%.

TABLE II Performance of models on the IMDb sentiment classification

Model	**Labeled data in IMDb training data**				
	0.5%	***1%***	***2%***	***4%***	***100%***
LSTM	66.30%	71.64%	73.81%	80.95%	82.38%
W2V LN	73.82%	77.12%	80.24%	81.48%	85.31%
BERT base LN	71.40%	75.19%	78.14%	78.13%	83.52%
BERT large LN	74.20%	76.26%	77.54%	79.24%	86.13%
DistilBERT LN	75.52%	79.20%	80.91%	81.85%	85.68%
ALBERT LN	**76.69%**	**79.43%**	**81.49%**	**83.36%**	**88.24%**

TABLE II includes the main results of our experiments. LSTM suffers more by reducing the amount of labeled training data. This is because the core part in LSTM called forget gate depends on old state of output gate. And the lacking of labeled data leads to undertraining on the weight of forget gate. In Ladder network, after 4-layer Noise encoder unlabeled data will get an available labeled. To combine these relabeled data with the few labeled data LN can give a better performance, especially on DistilBERT and ALBERT.

TABLE III Performance of models on the IMDb[1] and YelpNYC[2]

Dataset Model	**Labeled data in training data**				
	0.5%	***1%***	***2%***	***4%***	***100%***
DistilBERT[1]	75.52%	79.20%	80.91%	81.85%	85.68%
ALBERT[1]	**76.69%**	**79.43%**	**81.49%**	**83.36%**	**88.24%**
DistilBERT[2]	**46.19%**	47.49%	48.77%	**50.47%**	55.85%
ALBERT[2]	45.96%	**47.90%**	**49.32%**	49.69%	**57.99%**

In TABLE III, with the labeled data decreasing LN models can keep an effective classification both on binary classification and 5-class classifications.

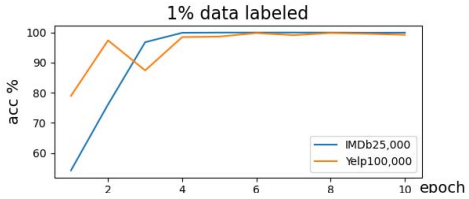

Figure 4. Comparison of accuracy changing on training data

Figure 4 is the accuracy of each epoch on the IMDb and YelpNYC dataset. We can see from Figure 4 that the model accuracy becomes stable in 5 epoch no matter which actual scene to face.

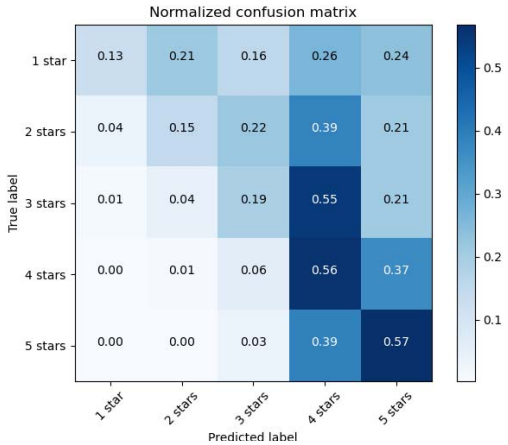

Figure 5. Confusion matrix on 1% labeled YelpNYC data with the ALBERT Ladder network

Figure 5 shows the confusion matrix about the IMDb dataset with the ALBERT-LN model with 1% data labeled in training. Furthermore, with the continuous sentiment

classification, our model has an error on the neighbor label 3 stars and 4 stars. Also, in a low number of stars, it has a misclassification. Our model considers hits review in 4 stars, though it is 1 star in the true label. Here is the text content "I loved this place. My partner and I had our first date here, and then our 6 month anniversary. It's super cute and scrumptious. And then I went in for lunch earlier today. When I asked the server how his day was going he scowled and said 'terrible'. It went down hill from there, and I walked out rather than try to enjoy myself under is hateful glare. I won't be returning." We can see in the first half of the review there are a large number of positive words. But the second half tells a bad experience in the restaurant with few negative words which lead the model wrong judgment. This is also a challenges we should look forward to solving in future.

D. Comparision with other learning methods

We compare our method with three traditional machine learning baselines: Naïve Bayes, Decision Tree, and Support Vector Machine (SVM) on two datasets in TABLE IV.

TABLE IV Machine learning baseline

Datasets	***Naïve Bayes***	***Decision Tree***	***SVM***
IMDb	65.97%	60.10%	**78.28%**
YelpNYC	33.42%	37.56%	**45.03%**

We selected 1% labeled data on the original training set. We also compared our method with LSTM as a neural-network-based learning model, and results are shown in TABLE II and V.

TABLE V Elapsed time for each neural-network-based learning model

Elapsed time	**Model**				
	W2V LN	***BERT LN***	***DISTBERT LN***	***ALBERT LN***	***LSTM***
IMDb	409.273s	397.994s	487.091s	391.834s	**2.070s**

We compared the elapsed time in each model to observe the time costs. And list as TABLE V. With 2-layer LSTM and one activation layer LSTM model get a short elapsed time on training.

V. CONCLUSION

In this paper, we focused on the problem of a few volumes of labeled training data and presented an approach for leveraging contextual features from unlabeled movie and restaurant reviews with a neural-network-based learning model, Ladder network. The experimental results showed that the method is effective for sentiment analysis, especially we verified that it works well on DistilBERT and ALBERT. Future work will include: (i) Scaling our approach to larger datasets like Peer reviews dataset, (ii) changing sentence level feature into word level and (iii) applying in aspect based sentiment analysis.

VI. ACKNOWLEDGEMENTS

We would like to thank Dr. Shebuti Rayana and Prof. Leman Akogle who provided us YelpNYC dataset. This work was supported by JSPS, Grant Number 18K11429.

REFERENCES

[1] Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks. In Advances in neural information processing systems (pp. 3546-3554).

[2] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies (pp. 142-150).

[3] Rayana, S., & Akoglu, L. (2015, August). Collective opinion spam detection: Bridging review networks and metadata. In Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining (pp. 985-994).

[4] Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture (pp. 70-77).

[5] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

[6] Esuli, A., & Sebastiani, F. (2006, May). Sentiwordnet: A publicly available lexical resource for opinion mining. In LREC (Vol. 6, pp. 417-422).

[7] Khan, F. H., Qamar, U., & Bashir, S. (2017). A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. Knowledge and information Systems, 51(3), 851-872

[8] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008, July). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning (pp. 1096-1103).

[9] Pezeshki, M., Fan, L., Brakel, P., Courville, A., & Bengio, Y. (2016, June). Deconstructing the ladder network architecture. In International conference on machine learning (pp. 2368-2376).

[10] Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In Advances in neural information processing systems (pp. 3079-3087).

[11] Nagesh, A., & Surdeanu, M. (2018, June). Keep your bearings: Lightly-supervised information extraction with ladder networks that avoids semantic drift. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (pp. 352-358).

[12] Dubey, H., Emmanouilidou, D., & Tashev, I. J. (2019, August). Cure Dataset: Ladder Networks for Audio Event Classification. In 2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) (pp. 1-6). IEEE..

[13] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).

[14] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[15] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[16] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[17] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942