Computer Science and Information Systems 18(4):1159–1174 https://doi.org/10.2298/CSIS201228044Y

Deep Semi-supervised Learning with Weight Map for Review Helpfulness Prediction

Hua Yin[1,⋆], Zhensheng Hu[1], Yahui Peng[2], Zhijian Wang[1], Guanglong Xu[3], and Yanfang Xu[4]

[1] Information School, Guangdong University of Finance & Economics,
Guangzhou, Guangdong, 510320, China
yinhua@whu.edu.cn
huzhsh6@mail2.sysu.edu.cn
1632646684@qq.com
[2] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou,
Guangdong, 510275, China
13924006758@139.com
[3] School of Statistics and Mathematics, Guangdong University of Finance & Economics,
Guangzhou, Guangdong, 510320, China
guanglongxu2018@gmail.com
[4] School of Art and Design, Guangdong University of Finance & Economics, Guangzhou,
Guangdong, 510320, China
kate20@student.gdufe.edu.cn

Abstract. Helpful online product reviews, which include massive information, have large impacts on customers' purchasing decisions. In most of e-commerce platforms, the helpfulness of reviews are decided by the votes from other customers. Making full use of these reviews with votes has enormous commercial value, especially in product recommendation. It drives researchers to study the technologies about how to evaluate the review helpfulness automatically. Although Deep Neural Network(DNN), learning from the historical reviews and labels, computed by the votes, has demonstrated effective results, it still has suffered insufficient labeled reviews problem. When the helpfulness of a large number of reviews is unknown for lack of votes, or some useful latest reviews with less votes are submerged by the past reviews, the accuracy of current DNN model decreases quickly. Therefore, we propose an end-to-end deep semi-supervised learning model with weight map, which makes full use of the unlabeled reviews. The training process in this model is divided into three stages:obtaining base classifier by less labeled reviews, iteratively applying weight map strategy on large unlabeled reviews to obtain pseudo-labeled reviews, training on above combined reviews to obtain the re-training classifier. Based on this novel model, we develop an algorithm and conduct a series of experiments, on Amazon Review Dataset, from the aspects of the baseline neural network selection and the strategies comparisons, including two labeling and three weighting strategies. The experimental results demonstrate the effectiveness of our method on utilizing the unlabeled data. And our findings show that the model adopted batch labeling strategy and non-linear weight mapping method has achieved the best performance.

Keywords: Semi-supervised learning; Review helpfulness; Pseudo label; Weight map; Labeling strategy.

⋆ Corresponding author

1. Introduction

Online reviews of products, which are very important to costumers, can provide reference for their purchase decisions [1] [2]. But massive customer reviews on e-commerce websites, including plenty of descriptive, emotional texts with diversified expressions, may lead to information overloading [3]. To highlight the useful reviews, some platforms allow users to vote on their helpfulness. But these votes are imbalanced on some new or unpopular products. The fewer votes on latest products lead to bias errors and are not credible. Therefore, it is necessary to automatically evaluate the review helpfulness[4].

Early studies mainly used hand-crafted features to try to solve the problem of review helpfulness prediction, such as Geneva affect label coder (GALC)[5], linguistic inquiry and word count (LIWC), general inquirer (INQ)[6]. However, due to manual feature engineering and data annotation, using manual features is laborious and expensive. Recently, models built using the convolutional neural networks (CNN)[7] have been applied to review usefulness predictions and showed performance increasing on review helpfulness prediction.

However, most of the current models have poor generalization ability when there is insufficient data. For products with few reviews, it is difficult to obtain enough label data to train effective models using supervised learning methods. In order to alleviate the issues of insufficient labeled data and make full use of adequate unlabeled data, we studies semi-supervised learning for review helpfulness prediction. This paper proposes a deep semi-supervised learning method with weight map for review helpfulness prediction without any hand-craft features and prior knowledge.

The remainder of the paper is organized as follows. Section 2 analysed the related work, Section 3 formally defines the problem and presents our method step by step. Section 4 illustrates the experiment settings including dataset, evaluation standards, experiment design and the experiment results analysis. Finally, Section 5 discusses the conclusions and the future work.

2. Related work

Some pioneering works hypothesize that helpfulness is an internal property of review texts, and try to find new hand-crafted features to study it. Martin used GALC [5] to extract the emotion features from the review texts to build the emotion-based review helpfulness prediction model [8]. Yang leveraged existing linguistic and psychological dictionaries to represent reviews in semantic dimensions [6]. Liu used some argument-based features as the indicators of helpful reviews [9]. However, the performances of these methods depend largely on the hand-crafted features and a mass of manual annotated samples, which are time-consuming and labor-intensive.

Hence, some neural network-based methods have been proposed to solve this problem. Lee and Choeh used multi-layer perceptron neural networks with hand-crafted features [10], similar with the work of Malik and Hussain who used deep neural network with emotion features [11]. Chen and Yang designed a convolutional neural network(CNN) on the raw-text reviews without any hand-crafted features [12]. Saumya used a two-layered convolutional neural network model to predict the best helpful online product review [13]. The models they built showed performance increasing on review helpfulness prediction.

Most of the existing works focus on popular product categories with massive reviews. However, in the case of insufficient data, the model generalization ability is poor. For example, the '*Electronics*' category of the Amazon Review Dataset [14] has more than 1.68 million reviews, while the '*Musical Instruments*' category only has 10k reviews, and most of them have a few votes. For products with a few reviews, it is difficult to obtain enough labeled data to train an effective model with supervised learning method.

Semi-supervised learning [15,16] was prompted to alleviate the issues of insufficient labeled data and make full use of adequate unlabeled data. The most classic and simplest form of semi-supervised learning is self-training[17,18,19,20]. It is an iterative process, which firstly trains a supervised classifier on the labeled data, and utilizes this classifier to label the unlabeled data, then enlarges the training set with the most confident predictions (also named pseudo labels[21]). This method can improve classifier's performance, especially when the labeled training data is obviously scarce[22]. However, classification errors might be accumulated along the process when the pseudo labels are not predicted correctly. The essential problem of self-training is how to make the baseline classifier more precisely and decrease the impact of false predicted reviews in the training process.

This paper proposes a deep semi-supervised learning method with weight map to predict review helpfulness automatically.The contributions of this paper are as follows:

1) The method is a new end-to-end self-training model for review helpfulness predictions, and the performance is considerable well in insufficient labeled reviews situation.
2) It proposes a novel deep semi-supervised learning framework with different labeling and weight mapping strategies,which guides the model to choose more reliable pseudo labels.
3) It develops an algorithm and conducts a series of experiments from the aspects of baseline and strategies comparisons on Amazon Review Dataset. The experimental results demonstrate the effectiveness of our method on ultilizing the unlabeled data.

3. Methodology

In this paper, A deep semi-supervised learning method is proposed for review helpfulness prediction. The flowchart of the proposed method is shown in Figure 1.The procedure is divided into three phases, including: (1)Training base classifier. (2) Weighting unlabeled reviews. (3) Re-training classifier. First, A base classifier is trained on the small labeled dataset by deep neural network. Second,the large unlabeled dataset is predicted by the base classifier for getting the pseudo labeled dataset. A probability selection is applied on this pseudo labeled dataset to get the selected labeled dataset. To balance the instances, a weighting map is proposed and applied on the selected labeled dataset. Then the weighted labeled dataset is combined with the original small labeled dataset to construct the re-train labeled dataset. Finally, the classifier is built on the expanded dataset. The details of each step are described in the following sections.

3.1. Preliminary

Given a small labeled review dataset $D_l = \{(x_i, y_i)|i = 1, 2, \ldots, n\}$ including n reviews, where x_i represents the ith user review and y_i represents the label of this review. The

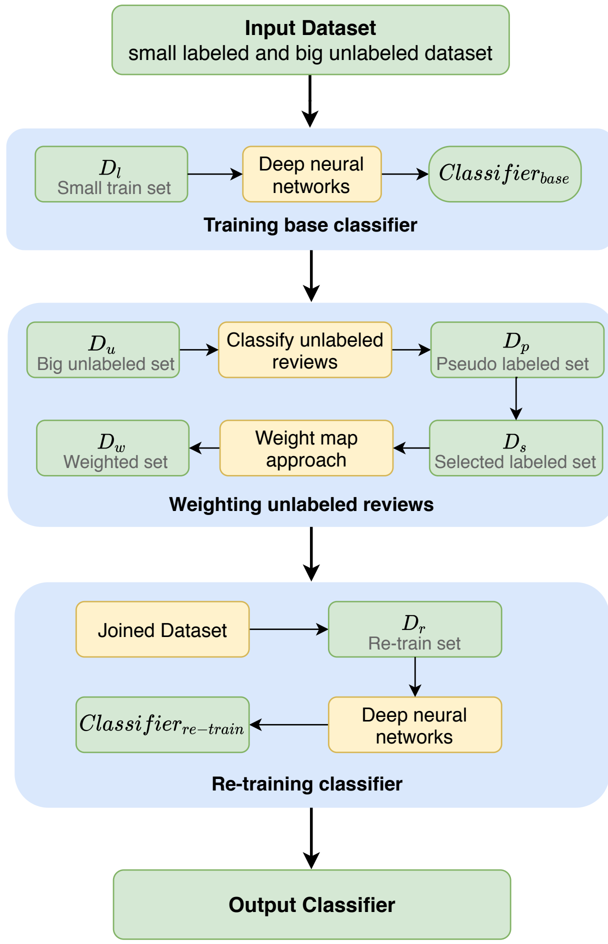

Fig. 1. The flowchart of the method

value of y can only be 0 or 1. When $y_i = 1$, it means that the review is helpful, otherwise it is unhelpful. $D_u = \{(x_j)|j = 1, 2, \ldots, m\}$ is the unlabeled large dataset including m reviews, where $n \ll m$. The review helpfulness prediction problem is defined as a binary classification problem to output a classifier which makes full use of both of the two datasets.

According to the flowchart, the pseudo labeled dataset $D_p = \{(x_j, y'_j, p_j)|j = 1, 2, \ldots, m\}$, where p_j represents the probability of $y'_j(y'_j = 0 or 1)$, is firstly obtained by the base review classifier built on D_l . Based on p_j, parts of reviews from D_p are selected to get $D_s = \{(x_j, y'_j, p_j)|j = 1, 2, \ldots, m'; m' < m\}$, where $D_s \subset D_p$. The rest reviews of D_p is D'_u. After D_s is processed by weighting map approach, the weight set $D_w = \{(x_j, y'_j, w_j)|j = 1, 2, \ldots, m'; w \in (0, 1]; m' < m\}$ is produced, where w_j is the weight of x_j. Then let $D_r = \{(x_k, y'_k, w_k)|k = 1, 2, \ldots, n + m'; w \in (0, 1]\}$ be the

Re-train set, which combined D_l with D_s. The symbol descriptions are showed in Table 1.

Table 1. Symbol Descriptions

Symbol	Description
D_l	the original small labeled training set with n reviews
D_u	the larger unlabeled training set with m reviews
D_p	pseudo labeled training set on D_u with m reviews
D_s	selected pseudo labeled training set on D_p with m' reviews
D_w	weighted pseudo labeled training set on D_s with m' reviews
D_r	new training set after combined D_l with D_w

3.2. Processing unlabeled reviews

In the training base classifier phrase, the base classifier $Classifier_{base}$ is built by the deep neural networks, not limited to CNN [7], Gated CNN [23,24,25] and RNN [26] to minimize the cross entropy error function on D_l. For the pretraining language models have demonstrated the state-of-the-art performance on a wide range of natural language processing tasks [27], BERT[28] is applied as the word embedding layer. As a key phrase, processing the unlabeled reviews is divided into three steps.

1) Generate the pseudo labeled set

The architecture of the model with deep neural networks are shown in figure 2. Given an unlabeled review x_j in D_u, we predict this review's label y'_j by $classifier_{base}$. As a binary classification problem, the output of the $Classifier_{base}$ is a two-dimensional vector o_{jk}, where $k = 0, 1$,for x_j. After training on D_u, we get an output matrix $Output$. For weighting process, we transfer matrix $Output$ to a probability matrix $Prob_{Output}$ computed by formula (1).

$$Output = \begin{bmatrix} o_{10} & o_{11} \\ o_{j0} & o_{j1} \\ \vdots & \vdots \\ o_{m0} & o_{m1} \end{bmatrix}$$

$$Prob_{Output} = \begin{bmatrix} p_{10} & p_{11} \\ p_{j0} & p_{j1} \\ \vdots & \vdots \\ p_{m0} & p_{m1} \end{bmatrix}$$

$$\begin{aligned} p_{jk} &= p(y'_j = k|x_j) \\ &= \frac{e^{o_{jk}}}{\sum_{i=0}^{i} e^{o_{ji}}} \end{aligned} \tag{1}$$

$$\text{where } k = 0, 1 \; i = 1, \; \sum p_{jk} = 1.$$

Then we obtain the pseudo label y'_j and p_j for each unlabeled review x_j in D_u by formula (2), (3).

$$y'_j = \begin{cases} 1, & if\ p_{jk} > 0.5, k = 1 \\ 0, & if\ p_{jk} \geq 0.5, k = 0 \end{cases} \tag{2}$$

$$p_j = p(p_{jk} | k = y'_j) \tag{3}$$

2) Select the pseudo label set

It is an iteration process to select the reviews from D_p to get D_s. We firstly choose the reviews whose p_j is larger than the mean of the p_j. These reviews construct D_s, and the rest of reviews in D_p construct D'_u,which is used in the next iteration.

$$p_{mean} = \frac{\sum_{j=1}^{m'} p_j}{m'} \tag{4}$$

3) Generate the weighted set

In the following retraining step, we combine the new labeled dataset with the original labeled dataset D_l to produce a new classifier. There are two ways to use the labeled dataset D_s. One is that all the reviews in D_s are treated as the same confidence,and another one is to treat them as different weights. Therefore, we set a weight factor on D_s, and transfer D_s to D_w. w_j is generated by formula (5).

$$w_j = \begin{cases} 1, & \text{No weight} \\ p_j, & \text{Hard weight} \\ f(p_j), & \text{Mapping weight} \end{cases} \tag{5}$$

When w_j is set to 1, D_s is transferred to D_w as following.

$$D_w = \left[\begin{array}{cc|c} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_{m'} & y'_{m'} & 1 \end{array} \right]$$

The hard weight utilizes the p_j in D_s. It sets p_j as the weight.

$$D_w = \left[\begin{array}{cc|c} x_1 & y_1 & p_1 \\ x_2 & y_2 & p_2 \\ \vdots & \vdots & \vdots \\ x_{m'} & y'_{m'} & p_{m'} \end{array} \right]$$

We propose a new soft weight mapping approach. It adjusts the weight based on hard weight to a more reasonable way.

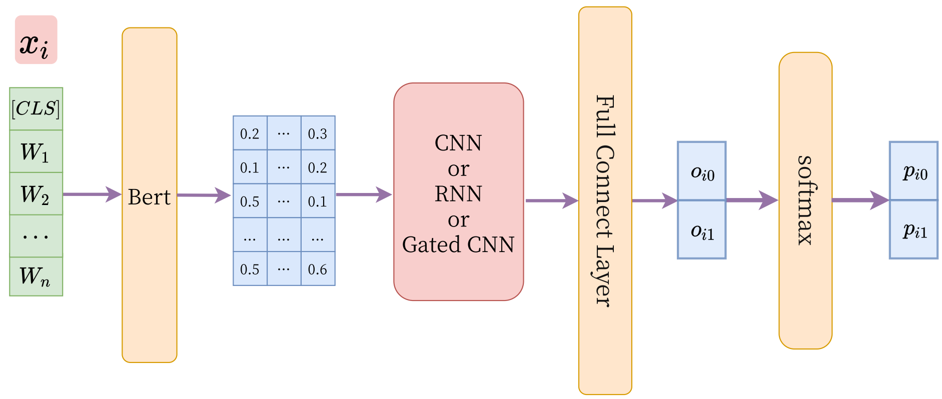

Fig. 2. The architecture of deep neural networks

3.3. Weight mapping approach

For the lower value of p_j, its influence should be lowered to the future classification model. On the other hand, more attention should be paid to the instances which have higher value of p_j. Weight mapping works.

Before being mapped, the range of p_j is $[p_{mean}, p_{max}]$. The confidence of each instance is very close. So the paper tries to map the original p_j to the range of $[0.5, 1]$. Linear mapping and non-linear mapping are two different mapping way to be compared. The linear mapping is done by formula (6). This mapping way doesn't change p_j's distribution density showed as Figure 3. The non-linear mapping is done by formula (7). The aim range is still $[0.5, 1]$. From Figure 3, it is found that non-linear mapping makes the instances diffuse to the side way and changes the original distribution. It makes more instances in the two sides of the new distribution.

$$w_j = \frac{(p_j - p_{mean}) * 0.5}{p_{max} - p_{mean}} + 0.5 \tag{6}$$

$$w_j = \min(\max(p_j(p_j - \frac{p_{max} - p_{mean}}{2} + 1), 0.5), 1) \tag{7}$$

3.4. Re-training classifier

Retraining phrase, shown as figure 4, is a semi-supervised learning process, which iterative utilize the processed unlabeled reviews. It includes three parts:join datasets, retrain classifier and terminate training.

The paper combines D_w with D_l to get a new dataset D_r,which has $n + m'$ reviews. For the reviews in D_l, its weight w_k is set to 1. Then the neural network is trained on the combined dataset D_r and the new classifier $Classifier_{re-train}$ is gotten. L_{ce} is the cross entropy loss function.

$$L(D_r) = L_{ce}(w_k \odot x_k, y_k) \tag{8}$$

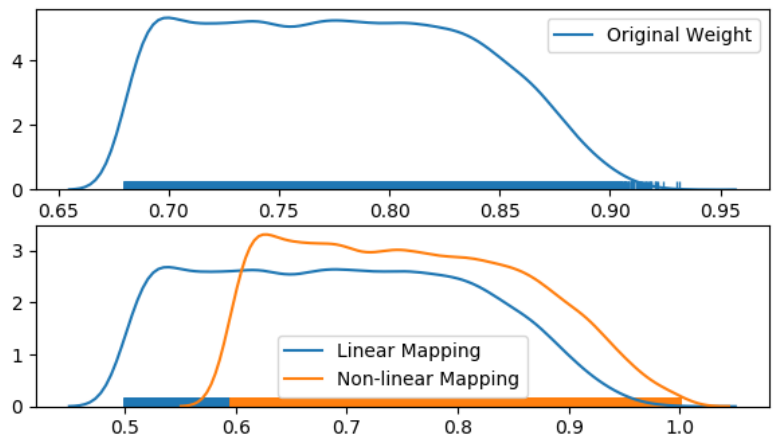

Fig. 3. Linear Mapping and Non-linear Mapping

$$L_{ce}(x_k, y_k) = -[y_k \cdot log(p_k) + (1 - y_k) \cdot log(1 - p_k)] \quad (9)$$

Retraining on the D_r is an iterative process. It will be terminated and output the final classifier when the accuracy of classifier does not increase any more.Or the number of reviews in D'_u is smaller than in D_l, then the training process will be terminated.

4. Experiment

In this part, the detailed experiments,including dataset processing, evaluation metrics, experiment settings and results analysis,are illustrated.

4.1. Dataset

The benchmark dataset is from Amazon Product Review Dataset. It has product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014 [14,29]. The reviews information contains ratings of products, texts of reviews, helpfulness votes and total votes of reviews. The paper chooses Electronics as a representative category to verify the proposed method. This category has the most reviews and is the most frequently used product category in related work.

In order to avoid data bias, the reviews with a total of less than 6 votes are removed, the proportion of helpful votes and unhelpful votes is set to 0.5 [30]. The paper randomly selects some of the helpful reviews to make the training dataset distribution to reach a balanced state, that is, the reviews marked as helpful and unhelpful are both half of the data set. To satisfy with the data setting requirement which the number of D_l is largely smaller than D_u, 1% of the original dataset is selected as the training dataset [20], and 20% of

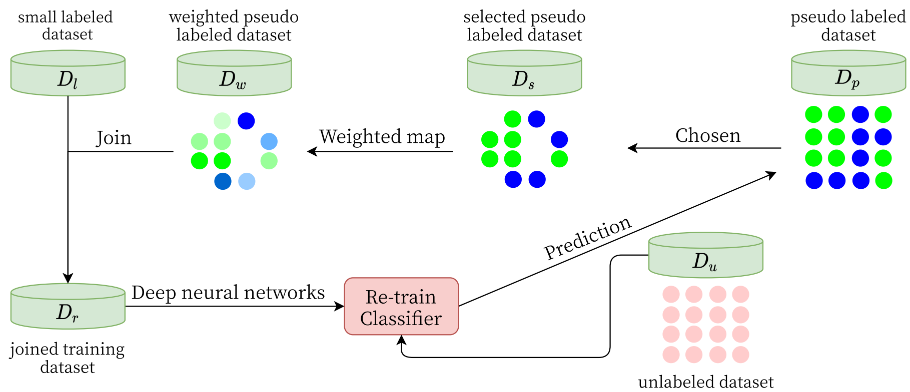

Fig. 4. Process of re-training classifier

the trainning dataset as the development set, and 10 times of the trainning dataset as the test set. The left of the original dataset is the unlabeled dataset. The detailed description is shown in Table 2.

Table 2. Datset Divisions

	Reviews	Tokens(unique)	Tokens
Train Set	500	10670	59378
Dev set	100	3585	11723
Test set	5000	54592	654264
Unlabeled	42934	232729	4789641

4.2. Evaluation metrics

Accuracy, Precision, Recall, F1-Score are chosen as the performance measures for evaluating the classification performance of our approach. The values of these evaluation criteria range from 0 to 1. The larger of these evaluation criteria, the performance of the model better.

Accuracy refers to the proportion of reviews correctly classified by the model. The calculation method is ass:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

F1-score considers both the Precision(P), which refers that the proportion of helpful reviews identifications is actually correct, and the Recall(R), of the test to compute the score, which are defined as:

$$
\begin{aligned}
F_1\text{-}score &= 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \\
Precision &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN}
\end{aligned}
\tag{11}
$$

Where a true positive (TP) is an outcome where the model correctly predicts the helpful reviews. Similarly, a true negative(TN) is an outcome where the model correctly predicts the unhelpful review. A false positive(FP) is an outcome where the model incorrectly predicts the helpful reviews. And a false negative(FN) is an outcome where the model incorrectly predicts the unhelpful review.

4.3. Experiment setting

To validate the efficiency of the deep semi-supervised learning method, it designs the experiments to answer the three important questions. Has the different deep neural network impacted on the final classifier? Which is the best choice in the three weight mapping approaches? What are the influences of the two labeling strategies?

1) Baseline setting

The paper chooses three typical deep neural network as the baseline network including Convolutional Neural Networks (CNN) [7], Gated CNN [23,24,25] and Recurrent Neural Networks(RNN) [26] , and sets the parameters of these three network to select better-performing network. For the Convolutional Neural Networks, the model consists of one convolutional layer with the 256 channels. The paper adopts multiple sizes of kernels 2, 3, 4, followed by ReLU activation [31]. It sets dropout rate to 0.3 for regularization [32], and concatenates them after every max-pooling layer, then trains the model using AdamW optimizer[33] with 1e-4 learning rate.The setting of model with Gated CNN is same as [23,24], and the model with RNN mainly refers to [26]. The word embedding model used in all experiments in this paper is Bert [28], and the max length of the review text used is 420, covering 95% of the effective review data. All the experiments are conducted on NVIDIA GeForce GTX 2080Ti GPU and implemented using PyTorch.During the training process, due to that the number of training data is very small, the early-stopping strategy [34] is adopted to prevent the model from overfitting.

2) Strategy comparison experiments description

The comparative experiments mainly verify the relevant strategies proposed above. One is a comparison of labeling strategies, and the other is a comparison of different weight mapping strategies.Labeling strategies includes overall labeling and batch labeling strategy. The overall labeling strategy means that the unlabeled set Du is used as a whole for prediction processing, and the remaining ones are filtered out and iteratively predicts the labeling process again. In contrast, the batch labeling strategy is to divide Du into batches in advance into $Du_1, Du_2, \ldots$, and then predict and filter each subset. Experiments will be conducted to analyze the impact of different labeling strategies. The weighting strategies includes three different strategies:no weight, hard weight and mapping weight. The effects of different weight strategies will be compared by experiments.

The table 3 is an explanation of the models and related strategies built by each comparative experiment. The neural network used in all experiments is the one performing well in the above baseline experiment.

Table 3. Comparative experiments description

Experiments	Weight Method	Labeled Method
Exp.1	No weight	Overall labeling
Exp.2	No weight	Batch labeling
Exp.3	Hard weight	Overall labeling
Exp.4	Hard weight	Batch labeling
Exp.5	Liner mapping weight	Overall labeling
Exp.6	Liner mapping weight	Batch labeling
Exp.7	Non-liner mapping weight	Overall labeling
Exp.8	Non-liner mapping weight	Batch labeling

4.4. Results and Analysis

After getting the results of baseline experiments, the better neural network was selected, then we analysed the influences of the two labeling strategies and the final results of the experiments.

1) Baseline experiments results

In order to select a better neural network, Three benchmark experiments have been conducted. The results of these experiments are shown in the following table 4. Based on the results, it can be concluded that the CNN network is generally better than the others from values of F1-score and accuracy.

Table 4. Baseline experiments Results

Model	Precision	Recall	F1-score	Accuracy
CNN	68.12	67.21	**67.66**	**67.33**
Gated CNN	67.51	66.32	66.91	67.10
RNN	63.64	70.02	66.67	65.11

2) Strategies experiments Analysis

When using the overall labeling strategy, in each training loop, the number of pseudo-labeled samples reduce and its F1-score is as follows Figure 5. It can be shown from Figure 5 that when the overall labeling strategy is adopted, the amount of pseudo-labeled samples added is no more than half of the previous loop. During the process of training model, the pseudo labeled samples introduced become less and less. It causes more errors accumulated in the early stage and even leads to the model performance degradation.

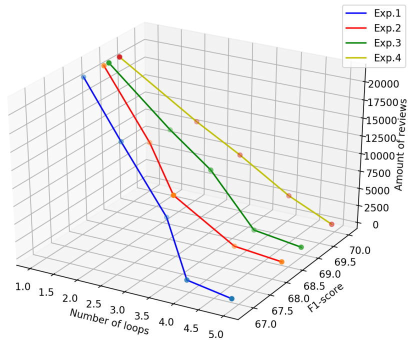

Fig. 5. F1-score and amount of reviews changed by loops on overall labeling strategy

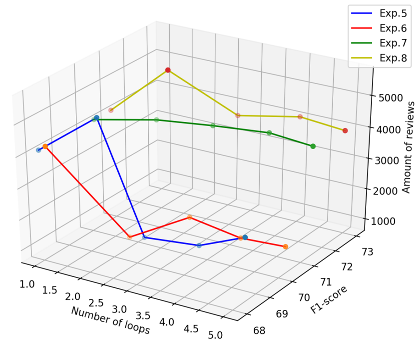

Fig. 6. F1-score and amount of reviews changed by loops on batch labeling strategy

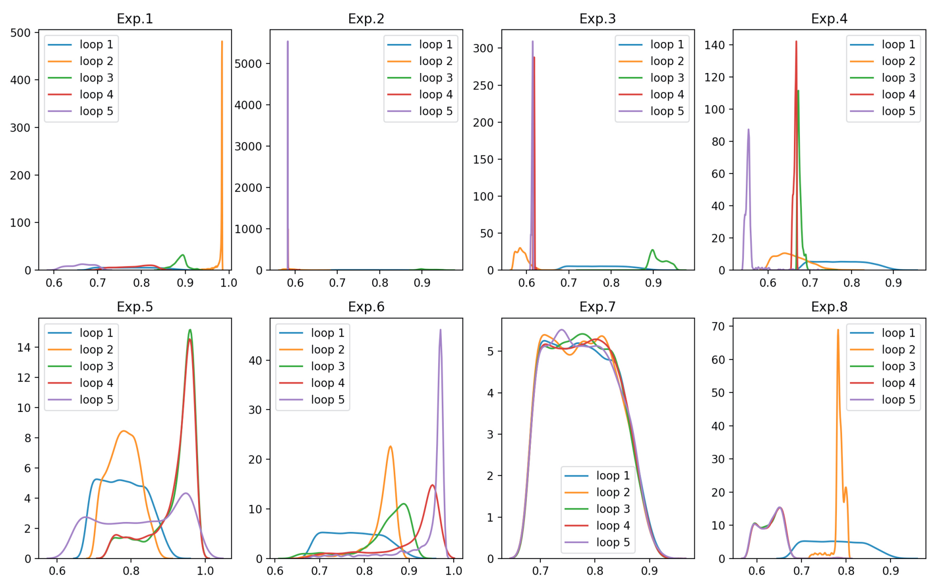

Fig. 7. Distribution density of p_j by loops (Exp.1-8)

When using the batch labeling strategy, the amount of pseudo-labeled reviews in each training loop and its F1-score are shown in the figure 6.

It can be demonstrated from Figure 6 that the experiments that only adopting batch labeling strategy without weight mapping processing will filter out more pseudo-labeled samples in the first loop. In the subsequent loops, the amount of reviews are slightly reduced, but remain stable. At the same time, experiments using batch labeling strategy and weight mapping method can prevent the model from a sharp decrease in the amount of pseudo-labeled training reviews, and make the amount of newly added pseudo-labeled reviews more stable in the entire training process. Thus the model performance can be stably improved.

The distribution and changes of the pseudo-labeled sample probability p_j in each semi-supervised training loop of Exp.1-8 are shown in the figure 7. When overall labeling strategy is used, in the semi-supervised training loop, the amount of pseudo-labeled training set added to the training loop decreases sharply. It results in a very large change in the probability distribution, which is extremely centralized. The stability and generalization of the model are both not enough. When the batch labeling strategy is adopted, the overall performance and improvement of the model are more stable, because the amount of pseudo-labeled samples added is relatively stable, and the model is more robust with a better generalization performance.

3) Analysis of the final results of experiments

The final results of the experiment are shown as table 5. It can be concluded that the model adopted batch labeling strategy and non-linear weight mapping method has the best experimental results. It's F1-score increases by 5.27% and accuracy increases by 4.96%, compared with the baseline model, which demonstrates obvious improvement.

Table 5. Final results of the experiment

Model	Precision	Recall	F1-score	Accuracy
Exp.1	67.82	68.53	68.17	68.06
Exp.2	66.39	71.26	68.73	68.27
Exp.3	67.21	71.01	69.05	67.93
Exp.4	71.25	68.91	70.06	69.42
Exp.5	69.46	67.63	68.53	68.34
Exp.6	69.23	71.80	70.47	70.48
Exp.7	71.12	72.21	71.66	71.21
Exp.8	70.12	76.22	**73.03**	**72.29**

5. Conclusion and future works

This paper studies semi-supervised learning for review helpfulness prediction. It proposes a deep semi-supervised learning method with weight map for review helpfulness prediction without any hand-craft features and prior knowledge. As the experiments demonstrated,the batch labeling strategy can effectively alleviate the problem of the sharp decrease in the pseudo-labeled sample size and make the pseudo-labeled data set flattened in the semi-supervised learning loop, and the weight mapping strategy can effectively improve the model effect, the stability and generalization of the model. In the future work, we will further explore the method of adjusting the pseudo-labeled sample weight in the semi-supervised learning process, and the application of semi-supervised learning in text classification.

Acknowledgments. This work was supported by Science and technology program of Guangzhou (No. 201707010495), the project of philosophy and social science planning of Guangzhou

(No. 2018GZYB77), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (No. 21YJCZH202) and the major project of basic and applied research in Guangdong universities (No. 2017WZDXM012).

References

1. Shuiqing Yang, Jianrong Yao, Atika Qazi, et al. Does the review deserve more helpfulness when its title resembles the content? locating helpful reviews by text mining. *Information Processing & Management*, 57(2):102179, 2020.
2. Juheng Zhang and Selwyn Piramuthu. Product recommendation with latent review topics. *Information Systems Frontiers*, 20(3):617–625, 2018.
3. Muhammad Shahid Iqbal Malik. Predicting users' review helpfulness: the role of significant review and reviewer characteristics. *Soft Computing*, pages 1–16, 2020.
4. Xiaoru Qu, Zhao Li, Jialin Wang, Zhipeng Zhang, Pengcheng Zou, Junxiao Jiang, Jiaming Huang, Rong Xiao, Ji Zhang, and Jun Gao. Category-aware graph neural networks for improving e-commerce review helpfulness prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2693–2700, 2020.
5. Klaus R Scherer. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729, 2005.
6. Yinfei Yang, Yaowei Yan, Minghui Qiu, and Forrest Bao. Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 38–44, 2015.
7. Yoon Kim. Convolutional neural networks for sentence classification, 2014.
8. Lionel Martin and Pearl Pu. Prediction of helpful reviews using emotions extraction. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, pages 1551–1557, 2014.
9. Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict and analyse review helpfulness. *arXiv preprint arXiv:1707.07279*, 2017.
10. Sangjae Lee and Joon Yeon Choeh. Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, 41(6):3041–3046, 2014.
11. MSI Malik and Ayyaz Hussain. Helpfulness of product reviews as a function of discrete positive and negative emotions. *Computers in Human Behavior*, 73:290–302, 2017.
12. Cen Chen, Yinfei Yang, Jun Zhou, and etc. Li. Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 602–607. Association for Computational Linguistics, June 2018.
13. Sunil Saumya, Jyoti Prakash Singh, and Yogesh K. Dwivedi. Predicting the helpfulness score of online reviews using convolutional neural network. *SOFT COMPUTING*, 24(15, SI):10989–11005, AUG 2020.
14. Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016.
15. Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.
16. Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

17. David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.
18. Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284, 2015.
19. Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3235–3246, 2018.
20. Hwiyeol Jo and Ceyda Cinarel. Delta-training: Simple semi-supervised text classification using pretrained word embeddings. *arXiv preprint arXiv:1901.07651*, 2019.
21. Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pages 1–6, 2013.
22. Heereen Shim, Stijn Luca, Dietwig Lowet, and Bart Vanrumste. Data augmentation and semi-supervised learning for deep neural networks-based text classifier. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 1119–1126, New York, NY, USA, 2020. Association for Computing Machinery.
23. Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941, 2017.
24. Cen Chen, Minghui Qiu, Yinfei Yang, Jun Zhou, Jun Huang, Xiaolong Li, and Forrest Bao. Review helpfulness prediction with embedding-gated cnn. *arXiv preprint arXiv:1808.09896*, 2018.
25. Cen Chen, Minghui Qiu, Yinfei Yang, Jun Zhou, Jun Huang, Xiaolong Li, and Forrest Sheng Bao. Multi-domain gated cnn for review helpfulness prediction. In *The World Wide Web Conference*, pages 2630–2636, 2019.
26. Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.
27. Zhensheng Hu, Hua Yin, Guanglong Xu, Yi Zhai, Danbei Pan, and Yongkang Liang. An empirical study on joint entities-relations extraction of chinese text based on bert. In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, pages 473–478, 2020.
28. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
29. Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.
30. Xianshan Qu, Xiaopeng Li, and John R Rose. Review helpfulness assessment based on convolutional neural network. *arXiv preprint arXiv:1808.09016*, 2018.
31. Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000.
32. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
33. Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *ICLR*, 2018.
34. Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

Hua Yin born in 1981. Associate Professor and Master supervisor in the Information School,Guangdong University of Finance and Economics. Her main research interests include machine learning, deep learning and natural language processing.

Zhensheng Hu born in 1991. Master student in the Information School, Guangdong University of Finance and Economics. His main research interests include deep learning and natural language processing.

Yahui Peng born in 1980. Ph.D.candidate in the School of Electronics and Information Technology, Sun Yat-sen University. His main research interests include data mining and medical image processing.

Zhijian Wang born in 1970. Professor and Master supervisor in the Information School, Guangdong University of Finance and Economics. His main research interestis business intelligence.

Guanglong Xu born in 1993. Master Student in the School of Statistics and Mathematics, Guangdong University of Finance and Economics. His main research interests include machine learning and natural language processing.

Yanfang Xu born in 1993. Master Student in the School of Art and Design, Guangdong University of Finance and Economics. Her main research interests include art design theory and data analysis and visualization.

Received: December 28, 2020; Accepted: May 05, 2021.