Heterogeneous-training: A Semi-supervised Text Classification Method

Yuhao Shen
Xihua University
Xinlan Xu
Xihua University
Bo Li
Xihua University
Bing Luo
Xihua University
Fei Hao
Shaanxi Normal University
Chao Zhang (galoiszhang@163.com)
Sichuan Police College

Research Article

Keywords: Text-classification, Machine Learning, Natural Language Processing, Ensemble Learning, Semi-supervised Learning

Posted Date: November 23rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2297963/v1

Additional Declarations: No competing interests reported.

ARTICLE TYPE

Heterogeneous-training: A Semi-supervised Text Classification Method†

Yuhao Shen[1] | Xinlan Xu[1] | Bo Li[1] | Bing Luo[1] | Fei Hao[2] | Chao Zhang*[3]

[1]School of Computer and Software Engineering, Xihua University, Sichuan, China
[2]School of Computer Science, Shaanxi Normal University, Shaanxi, China
[3]Intelligent Policing Key Laboratory of Sichuan Province, Sichuan Police College, Sichuan, China

Correspondence
*Chao Zhang, Intelligent Policing Key Laboratory of Sichuan Province, Sichuan Police College, Sichuan, China. Email: galoiszhang@163.com

Present Address
1. School of Computer and Software Engineering, Xihua University, 999 # Jin Zhou Rd. Jin niu District, Chengdu, Sichuan 610039, China.
2. School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi 710119, China
3. Intelligent Policing Key Laboratory of Sichuan Province, 186# Longtouguan Road, Jiangyang District, Luzhou City, Sichuan 646000, China.

Abstract

Using text classifiers, we can structure all manner of relevant text automatically, from emails, social media, surveys, legal documents, and more in a fast and cost-effective way. In this paper, a semi-supervised ensemble learning algorithm Heterogeneous-training is proposed and applied to the field of text classification. Based on the Tri-training algorithm, the Heterogeneous-training algorithm improves the traditional Tri-training algorithm by using different classifiers, dynamically updating the probability threshold and adaptively editing data. The experiments show that our method always outperforms Tri-training algorithm in text classification on benchmark text data sets.

KEYWORDS:
Text-classification · Machine Learning · Natural Language Processing · Ensemble Learning · Semi-supervised Learning

1 | INTRODUCTION

With the continuous development of computer technology and network technology, the text data on the network is increasing day by day. As the information carrier with the widest distribution and the largest amount of data, it is particularly important to scientifically organize and manage massive data by using text classification technology. So text classification is an important issue in the field of natural language processing in current machine learning field. Traditional supervised learning requires manual label of text categories, so the labor cost is high. Semi-supervised text classification can learn the information of unlabeled samples, so it attracts more and more attention. The most common weakly supervised method[1,2,3,4] of text classification involves generating pseudo-labels and training a classifier to learn the mapping between documents and classes. It is undeniable that pseudo-label quality contributes significantly to final classification accuracy. But because of their inspiration, they will inevitably produce noise. It is common for pseudo-labels to be generated by using some heuristic, such as String-Match between the documents and seed words[5] provided by the user. There is a high risk of erroneous predictions trained on such noisy labels. Therefore, it is necessary to reduce the noise generated by pseudo-labels as much as possible.

Ensemble learning[6], as a machine learning method, can well integrate multiple weak supervised classifiers, thus greatly reducing the noise. The first ensemble learning method: co-training algorithm was proposed by Blum et al[7]. In this method, it is assumed that the dataset has 2 sufficiently redundant views, each corresponding to one attribute set. There are 2 sufficient conditions for the definition of a view: each attribute set can describe the original dataset and each attribute set is independent of the other attribute set. However, in reality, these 2 conditions are difficult

†This is an example for title footnote.

to satisfy. For example, many data have multiple views, but not necessarily sufficient views[8]); the 2 views are not often sufficiently independent. The literature[9] does not require the problem itself to have sufficiently redundant views, but introduces restrictions on the types of classifiers and increases the computational overhead for estimating the labeling confidence.

In response to the shortcomings of Co-training, Tri-training[10] was developed to improve efficiency by avoiding the long validation time of traditional co-training while using multiple classifiers for co-training. In this algorithm, firstly, the labeled sample set is bootstrap sampling to obtain three labeled training sets, and then a classifier is generated from each training set. During the co-training process, the new labeled examples obtained by each classifier are provided by the other two classifiers in cooperation. Specifically, if the predictions of the two classifiers for the same unlabeled example are the same, the example is considered to have a high labeled confidence, and it is added to the labeled training set of the third classifier after being labeled. When predicting the unseen examples, Tri-training algorithm no longer selects one classifier to use like the previous algorithms, but uses the voting method often used in ensemble learning to form an ensemble of three classifiers to realize the prediction of the unseen examples. Unlike previous co-training algorithms, which need to explicitly estimate the confidence of labels, Tri-training algorithm implicitly compares the confidence of different unlabeled samples by judging the prediction consistency of three classifiers. Therefore, it is unnecessary to frequently use time-consuming statistical testing techniques.

The traditional Tri-training method implicitly compares the labeling confidence of different unlabeled examples by determining the prediction consistency of three classifiers.So it eliminates the need for frequent and time-consuming statistical testing techniques[11].The implicit treatment is often less accurate than the explicit estimation of labeling confidence.Especially if the initial classifier is weak, the unlabeled examples may be mislabeled, thus introducing noise into the training of the third classifier. In addition, although the algorithm defines an error constraint formula to constrain the introduction of noisy data, it does not consider the problem of unbalanced sample classes in the training set due to classifier error accumulation. Meanwhile, since the same one learning method to train the data, even if the training sets are different, the obtained classifiers have the same classification result for any data probability is greater than the probability of different.Affected by this, the generalization ability of Tri-training algorithm is not strong.

To address these problems, we propose the ideas of semi-supervised learning and co-training into text classification, and improves the traditional information gain algorithm and Tri-training algorithm, and proposes a semi-supervised learning algorithm based on the improved Tri-training algorithm. We propose a semi-supervised classification method Tri-training-New based on Tri-training algorithm to achieve text classification, and make full use of the unlabeled sample data to improve the classification performance.

The rest of this paper is mainly about the following contents: The second part explains our research on semi-supervised text classification, ensemble learning and Tri-training algorithm, the third part puts forward our own improved model, ANIESBNVJEHDNQAIBNIU, the fourth part writes our related experiments, and the fifth part expounds the relevant conclusions of this paper.

2 | RELATED WORK

2.1 | Semi-supervised Text Classification

There are many difficulties in text classification, the fundamental reason of which is various ambiguity or polysemy and evolution problems widely existing at all levels of natural language texts. For example, from the perspective of input length, texts are divided into short texts, long texts and super-long texts, and texts with different lengths are suitable for different classifiers. Secondly, from the label level, complex semantic recognition is another difficulty in text classification. Finally, many words have produced many new meanings over time, which makes text classification more difficult.

Recently, Semi-supervised Text Classification(SSTC) work has focused on deep self-training[12 13 14 15 16 17]. Classifiers can use marked and unmarked texts to learn water depth characteristics under a unified framework. This can be accomplished by performing a replacement process. In the replacement process, the current depth classifier updates the pseudo-labels of the unlabeled text and then retrains the depth classifier on marked and pseudo-labeled texts. For example, the virtual combat training (VAT) method[12 15] follows the principle of making the classifier resist random and local interference. First, it uses the current depth classifier to generate the prediction of the original text, then applies local perturbation to the embedding of the original text, and trains the depth classifier by using the consistency loss between the original prediction and the output of the depth classifier. Besides, the maximum likelihood confrontation training[14], virtual confrontation training and entropy minimization are combined into a unified goal. In addition, unsupervised data enhancement (UDA)[16] uses data enhancement techniques, such as reverse translation and tf-idf word substitution, to take advantage of the loss of consistency between the prediction of unlabeled text and the corresponding enhanced text instead of applying local perturbation. This work[13] makes use of cross-view training by matching the prediction of the auxiliary prediction module on the restricted view of unlabeled text (for example, only a part of a sentence) with the prediction of the main prediction module on the corresponding full view.

2.2 | Ensemble Learning

Ensemble learning[18] combines several independent models to get better generalization performance. Currently, deep learning models with multi-layer processing architecture are showing better performance as compared to shallow or traditional classification models.The main idea of ensemble learning is to generate multiple learners through certain rules, then combine them by some integration strategy, and finally make a comprehensive judgment and output the final result. Generally speaking, many learners in ensemble learning are homogeneous "weak learners". Based on this weak learner, several learners are generated through sample set disturbance, input feature disturbance, output representation disturbance and algorithm parameter disturbance, and a "strong learner"[19] with better accuracy can be obtained after fusion. With the deepening of integrated learning research, its broad definition[20] has been gradually accepted by scholars. It refers to adopting learning methods for multiple groups of learners without distinguishing the nature of learners. However, at present, the research on ensemble learning of homogeneous classifiers still accounts for the majority.

2.3 | Tri-training[10]

Many semi-supervised learning algorithms use the classifier generation model, and use expectation maximization to simulate the process of tag estimation or parameter estimation. A prominent achievement in this area is the co-training paradigm proposed by Blum and Mitchell[7], which trains two classifiers separately on two different views, I.E. Two independent sets of attributes, and uses the predictions of each classifier on unlabeled examples to augment the training set of the other. Dasgupta et al.[21] have shown that when the requirements are met, by maximizing their consistency on unlabeled data, jointly trained classifiers could produce fewer generalization errors. Unfortunately, in most cases, this demand is difficult to satisfy. Goldman and Zhou[9] proposed an algorithm that does not use attribute partition. However, it requires using two different supervised learning algorithms that partition the instance space into a set of equivalence classes, and employing the time-consuming cross-validation technique to determine how to label the unlabeled examples and how to produce the final hypothesis.

Zhou et al proposed a new co-training style algorithm, named tri-training. Tri-training does not need sufficient and redundant views, and does not need to use different supervised learning algorithms. It assumes that the instance space is divided into a group of equivalent categories. Therefore, it can easily be applied to common data mining scenarios. In contrast to previous algorithms that utilize two classifiers, tri-training uses three classifiers. This setting solves the problem of how to mark unlabeled examples and how to generate the final hypothesis, which makes a great contribution to the efficiency of the algorithm. In addition, combining these three classifiers can obtain better generalization ability.

The general procedure of the Tri-training classification algorithm is as follows. First, a small number of labeled data sets L are initialized, and three classifiers$\mathrm{H}_1, \mathrm{H}_2$,and $\mathrm{H}_3$are trained by L. If x is any point in the unlabeled data set U, andH_2 ,and H_3have the same classification result for x, then x is labeled as $\mathrm{H}_2(\mathrm{x})$ and added to the training set of H_1, thus forming a new training set $\mathrm{S}_1 = \mathrm{L} \cup \{\mathrm{x}/\mathrm{x} \in \mathrm{U} \mathrm{and}\, \mathrm{H}_2(\mathrm{x}) = \mathrm{H}_3(\mathrm{x})\}$ for H_1. Similarly, the training sets of H_2 and H_3 are expanded to S_2 and S_3, respectively. Tri-training compares the labeled confidence of different unlabeled samples instead of explicitly comparing the labeled confidence. This avoids the need for frequent and time-consuming statistical testing techniques. However, the downside is that this implicit process is often less accurate than the explicit estimation of labeling confidence, especially when the initial presentation classifier is weak and unlabeled samples may be mislabeled, thus introducing noise to the third classifier. labeled samples are accurate then the negative impact of the introduced noisy data can be offset by the benefit of using a large number of unlabeled samples.

3 | HETEROGENEOUS-TRAINING

Tri-training has achieved surprising results in classification by using semi-supervised ensemble learning. However, the Tri-training algorithm itself has some problems.The traditional Tri-training method implicitly compares the confidence of different unlabeled samples by determining the prediction consistency of three classifiers. Therefore, it eliminates the need for frequent and time consuming statistical testing techniques. The implicit processing is usually less accurate as the explicit estimation of the confidence of the mark. Specifically, if the initial classifier is weak, the unlabeled examples may be mistakenly labeled, thus introducing noise into the training of the third classifier. In addition, although the algorithm defines an error constraint formula to constrain the introduction of noise data, it does not consider the problem of sample class imbalance in the training set caused by the accumulation of classifier errors. At the same time, because the same learning method is adopted to train the data, even if the training sets are different, the probability that the classifier can get the same classification result for any data is greater than the probability of different classification results. Affected by this, the generalization ability of Tri-training algorithm is weak. Moreover, the new labeled examples obtained by each classifier are collaboratively provided by the remaining 2 classifiers during the co-training process[22 23]. The proportion of sample categories in the unlabeled data is unknown, and noise data is constantly introduced in the training process, which leads to the continuous accumulation of discrimination errors of classifiers and the imbalance of sample categories in the training set.

To address these problems, we proposes an improved Tri-training algorithm.Compared with the original Tri-training algorithm, our new algorithm has the following five improvements.

1. Use three different supervised classification algorithms as learning algorithms.
2. The proposed improved algorithm shares one training set among three classifiers.
3. More strict restrictions on samples entering labeled data sets.
4. Combining the RemoveOnly[24] editing operation and adaptive data editing strategy[25] into the Tri-training learning process.
5. Track the training the probability thresholds of the 3 classifiers are dynamically updated for the same unlabeled sample.

The steps of text classification include text preprocessing, text feature selection and classifier training. Our new model will be described from these three steps.At the same time,the flow and characteristics of the algorithm will be described in detail below.

3.1 | Text Preprocessing

Text preprocessing refers to the standardized processing of the selected text data so that it can be converted into a form for subsequent feature extraction, including word segmentation, stop words removal, special character deletion and root reduction of English words. Among them, word segmentation is the word obtained by using English spaces and punctuation marks as separators. To stop words is to eliminate meaningless words and filter them with commonly used English stoppages. Delete special characters Use regular expressions to filter other characters except English characters, numbers and punctuation; The simplification of English words roots is to obtain the basic form of words by removing affixes.

3.2 | Selection Of Text Features

Information gain is a common feature selection algorithm in text classification. Information gain reflects the importance of features, and the larger the information gain, the more important the features are.

The total information gain formula is as follows.

$$\mathrm{IG}(t)=-\sum_{i=1}^{k} P\left(C_{i}\right) \log_2 P\left(C_{i}\right)+P(t) \sum_{i=1}^{k} P\left(C_{i} \mid t\right) \times \log_2 P\left(C_{i} \mid t\right)+P(\bar{t}) \sum_{i=1}^{k} P\left(C_{i} \mid \bar{t}\right) \log_2 P\left(C_{i} \mid \bar{t}\right) \tag{1}$$

k is the number of patent types; $P\left(C_{i}\right)$ is the probability that a certain type of patent text appearing in the total number of all types of patent texts,$P\left(C_{i}\right)=N_{C_{i}}/N$. $N_{C_{i}}$ is the number of certain types of patent texts and N is the total number of patent texts; $P\left(t\right)$ is the probability that the text containing feature t appears in all patent text,$P\left(t\right)=N_{t}/N$, and $N_{t})$ is the number of patents containing feature t; $P\left(C_{i} \mid t\right)$ is the conditional probability that the patent belongs to class C_{i} when the feature t appears; $P\left(C_{i} \mid t\right)=N_{c_{i}\cap t}, N_{c_{i}\cap t}$ is the number of texts with feature t in the class C_{i}; $P(\bar{t})$ is the probability of the text without feature t appearing in the total patent text, $P(\bar{t})=N_{\bar{t}}/N, N_{\bar{t}}$, and $N_{\bar{t}}$ is the number of patents without feature t; $P\left(C_{i} \mid \bar{t}\right)$ is the conditional probability that the patent belongs to class C_{i} when feature does not appear, and $P\left(C_{i} \mid \bar{t}\right)=N_{c_{i}\cap\bar{t}}/N_{\bar{t}}, N_{c_{i}\cap\bar{t}}$ is the number of patents that do not have feature in class C_{i}.

3.3 | Classifier Training

Let D_{i} be a labeled balanced training set containing $|D_{i}|$ samples and a given set of labels $\mathrm{C}=\{c_{1}, c_{2}, c_{3}, \ldots, c_{m}\}$; D_{U} be an unlabeled training set containing $|D_{U}|$ samples:$\left\{d_{u_{1}}, d_{u_{2}}, \ldots, d_{u_{|D_{U}|}}\right\}$; and the validation and test sets be D_{v} and $D_{t}.D_{i}$,D_{v} and D_{t} obey the same data distribution. Let the three initial classifiers h_{2} ,h_{2} and h_{3} be trained by the original labeled balanced training set, and use these three classifiers to make prediction judgments on the unknown sample set, and only those samples that satisfy the aforementioned conditions (i. e. , all three classifiers have the same prediction for the same unlabeled sample, and the prediction probabilities given by all three classifiers are greater than their respective probability thresholds) can be added to the original training set. The prediction probabilities are calculated by the classification algorithms of the classifiers, and the set of $|C|$ probability distributions summing to 1 is obtained by all 3 classifiers for any sample $d_{u_{j}}$ in D_{U} , and the expression is:

$$P\left(d_{u_{j}}\right)=\left\{\mathrm{pro}_{h_{k}}^{c_{i}}\left(d_{u_{j}}\right) \mid 0<\mathrm{pro}_{h_{k}}^{c_{i}}\left(d_{u_{j}}\right)<1, k=1,2,3, d_{u_{j}} \in D_{U}, c_{i} \in C, \sum_{i=1}^{m} \mathrm{pro}_{h_{k}}^{c_{i}}\left(d_{u_{j}}\right)=1\right\} \tag{2}$$

Where the maximum probability is the predicted probability, namely

$$\max\left\{\mathrm{pro}_{\mathrm{h_k}}^{\mathrm{c_i}}\left(\mathrm{d_{u_j}}\right) \mid 0<\mathrm{pro}_{\mathrm{h_k}}^{\mathrm{c_i}}\left(\mathrm{d_{u_j}}\right)<1, \mathrm{c_i} \in \mathrm{C}\right\} \tag{3}$$

The probability thresholds are updated dynamically according to the changes in the proportion of sample categories after each iteration of training, so the initial set of probability thresholds for the 3 classifiers must be obtained first. The initial set of "pseudo-labels" predicted by the three classifiers for D_{U} is obtained by first using h_{1} , h_{2} and h_{3} to determine the unlabeled dataset D_{U} , and then using equation (4) to calculate the frequency of the number of labels of the three classifiers for m categories in L as The set of initial values of probability thresholds,which can be defined as:

$$\text{Per} = \left\{ \mathrm{per}_{\mathrm{h_k}}^{\mathrm{c_i}} \mid \mathrm{per}_{\mathrm{h_k}}^{\mathrm{c_i}} = \frac{\sum_{\mathrm{j=1}}^{|D_U|} \mathrm{F}\left(\mathrm{l_{h_k}}\left(\mathrm{d_{u_j}}\right) = \mathrm{c_i}\right)}{\sum_{\mathrm{i=1}}^{\mathrm{m=1}} \sum_{\mathrm{j=1}}^{|D_U|} \mathrm{F}\left(\mathrm{l_{h_k}}\left(\mathrm{d_{u_j}}\right) = \mathrm{c_i}\right)}, \mathrm{k} = 1,2,3, \mathrm{c_i} \in \mathrm{C} \right\} \tag{4}$$

Where $\mathrm{F}(\cdot) = 1$ when $\mathrm{l_{h_k}}\left(\mathrm{d_{u_j}}\right) = \mathrm{c_i}$, otherwise $\mathrm{F}(\cdot)$ is 0. The adaptive clipping strategy for RemoveOnly is as follows: let D^t represent round t as the newly labeled sample set of h_1 in round t-1 iteration, D_{de}^t as the remaining training set after clipping, and ε^t and ε^t as the hypothetical classification error rate of $D_i \cup D^t$ and $D_i \cup D_{de}^t$ for h_1 training more, respectively. If RemoveOnly has not been triggered in the $t-1^{th}$ iteration and $\left|D^t\right| > \left|D^{t-1}\right|$, RemoveOnly will be triggered under the guarantee of $\varepsilon_{de}^t < \varepsilon^t < \varepsilon^{t-1}$; if RemoveOnly has not been triggered in the $t-1^{\text{th}}$ iteration and $\left|D^t\right| > \left|D^{t-1}\right|$, RemoveOnly will be triggered if $\varepsilon^t < \varepsilon^{t-1}$ but $\varepsilon_{de}^t < \varepsilon^{t-1}$ cannot be satisfied; if RemoveOnly has been triggered in the $t-1^{\text{th}}$ iteration and $\left|D^t\right| > \left|D_{de}^{t-1}\right|$, h will be triggered under the guarantee of $\varepsilon_{de}^t < \varepsilon^t < \varepsilon^{t-1}$;if RemoveOnly has been triggered and $\left|D^t\right| > \left|D_{de}^{t-1}\right|$ in the $t-1^{\text{th}}$ iteration, RemoveOnly will be triggered if $\varepsilon^t < \varepsilon_{de}^{t-1}$ is not satisfied but $\varepsilon_{de}^t < \varepsilon_{de}^{t-1}$ is satisfied; except for the above cases, s will not be triggered.

Algorithm 1 explains the running process of the model. First, input original labeled balanced training set D_i , unlabeled training set D_U , validation set D_V test set D_t , number of subsets of unlabeled training set n, number of iterations t_0 , sample category proportion upper bound P_h , lower bound P_l , and fine-tuning step . Preprocess the labeled balanced training set. Reduce the dimension of the preprocessed ones, get the feature vector, and train the featur vector with Naive Bayes, SVM, and Xgboost(Chen et al. ,2016) to get three initial classifier h_1, $h_2$and h_3 with large differences. Use h_1, h_2 and h_3 to preliminarily determine D_U and get L^0. Calculate Per in L^0 for m categories.

$$Per = \left\{ \mathrm{per}_{\mathrm{h_k}}^{\mathrm{c_i}}, \mathrm{k} = 1,2,3, \mathrm{c_i} \in \mathrm{C} \right\} \tag{5}$$

Divide D_U into n subsets at random, use h_1, h_2 and h_3 to determine D_U , get D_{U_t}. $D_U = D_{U_1} \cup D_{U_2} \cup \ldots \cup D_{U_n}$, $D_{U_t} = \left\{ d_1^{U_t}, d_2^{U_t}, d_3^{U_t}, \ldots, d_{|D_{U_t}|}^{U_t} \right\}, \forall D_{U_t} \subset D_U$ .Use h_1, h_2 and h_3 to determine D_U , and get D_{U_t} :

$$\begin{gathered} \mathrm{L}\left(\mathrm{D_{U_t}}\right) = \left\{ \mathrm{ll_{h_k}}\left(\mathrm{d_1^{U_t}}\right) \mid \mathrm{ll_{h_k}}\left(\mathrm{d_1^{U_t}}\right) \in \mathrm{C}, \mathrm{k} = 1,2,3,\ \mathrm{d_1^{U_t}} \in \mathrm{D_{U_t}} \right\} \\ \mathrm{P}\left(\mathrm{D_{U_t}}\right) = \left\{ \mathrm{pro}_{\mathrm{h_k}}^{\mathrm{C_i}}\left(\mathrm{d_1^{U_t}}\right) \mid 0 < \mathrm{pro}_{\mathrm{h_k}}^{\mathrm{C_i}}\left(\mathrm{d_1^{U_t}}\right) < 1, \mathrm{k} = 1,2,3,\ \mathrm{d_1^{U_t}} \in D_U, \mathrm{c_i} \in \mathrm{C}, \sum_{\mathrm{i=1}}^{\mathrm{m}} \mathrm{pro}_{\mathrm{h_k}}^{\mathrm{C_i}}\left(\mathrm{d_1^{U_t}}\right) = 1 \right\} \end{gathered} \tag{6}$$

For each patent text $\mathrm{d_1^{U_t}}$ in D_{U_t}, $D_i += \left\{\mathrm{d_1^{U_t}}\right\}$, $D_U -= \left\{\mathrm{d_1^{U_t}}\right\}$ .The constraints are as follows:

$$\left\{ \begin{array}{c} \mathrm{ll_{h_1}}\left(\mathrm{d}_l^{\mathrm{U_t}}\right) = \mathrm{ll_{h_2}}\left(\mathrm{d}_l^{\mathrm{U_t}}\right) = \mathrm{ll_{h_3}}\left(\mathrm{d}_l^{\mathrm{U_t}}\right) = \mathrm{c_i}, \forall c_i \in C \\ \max\left\{ \mathrm{pro}_{\mathrm{h_k}}^{\mathrm{c_i}}\left(\mathrm{d}_l^{\mathrm{U_t}}\right) \mid 0 < \mathrm{pro}_{\mathrm{h_k}}^{\mathrm{c_i}}\left(\mathrm{d}_l^{\mathrm{U_t}}\right) < 1, \mathrm{c_i} \in \mathrm{C} \right\} \geq \mathrm{per}_{\mathrm{h_k}}^{\mathrm{c_i}}, \forall \mathrm{k} = 1,2,3 \end{array} \right. \tag{7}$$

If i (categories in D_i) $> P_h$, $per_{h_k}^{c_i} += step$, else $per_{h_k}^{c_i} -= step$.Then repeat until traverse through all the patent texts in D_{U_t} .Next, clip $\mathrm{D_i}$ with RemoveOnly and calculate the performance. Calculate two performance of RemoveOnly and determine whether RemoveOnly is trigger or inhibited according to that triggering condition in the adaptive strategy accord to the current situation, and for each h_1, h_2 and h_3, if it is triggered, updating the training set with the edited new marker sample set for retraining; if it is suppressed, the training set will still be updated according to the standard Tri-training method. After that, retrain h_1, h_2 and h_3 with updated $\mathrm{D_i}$; validate with $\mathrm{D_v}$,recording the result and iteration of at that time.Then,repeat until the number of iterations is equal to t_0 . Finally, take the best validation result among the 3 classifiers as the final classifier H, test it on D_t , and output test result F_1 .

4 | EXPERIMENTS

4.1 | Experimental Environment

The CPU is 5220R 2.2G, 24C/48T (24 core 48 process) * 2,server is EMC PowerEdge R740, GPU is NVIDIA GRX 3090, memory is 384G, operating system is Ubuntu 18.04, CUDA 11.4.0, Python 3.7 and Pytorch 1.2.

4.2 | Dataset

We conducted experiments on four datasets. The details of the dataset are as follows.

The New York Times (NYT): The NYT dataset is a collection of news articles published by The New York Times. They are classified into 5 coarse-grained genres (e. g. science, sports) and 25 fine-grained categories (e. g. music, football, dance, basketball).

20 Newsgroups (20 News): the 20 News dataset is a collection of newsgroup documents partitioned widely into 6 groups (e. g. recre-ation, computers) and 20 fine-grained classes (e. g. graphics, windows, baseball, hockey).

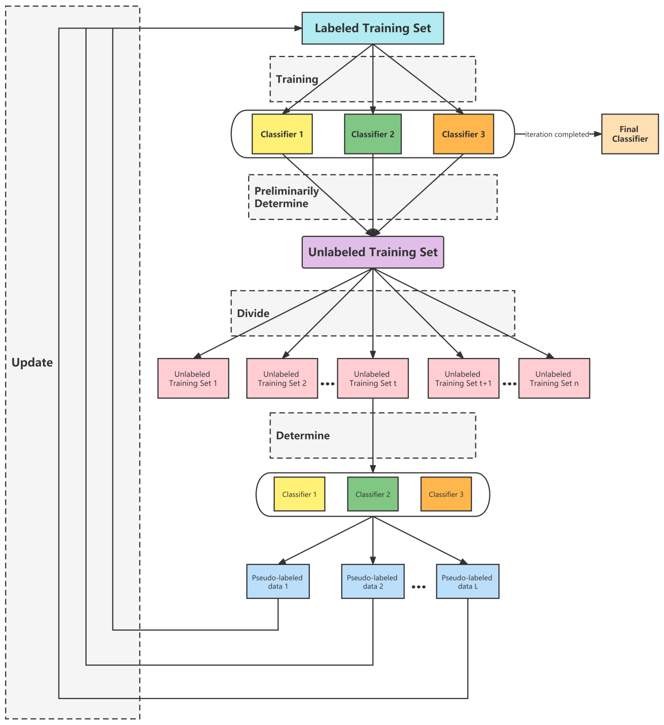

Figure 1 Schematic diagram of Heterogeneous-training algorithm model.In the figure, Labeled Training Set represents D_i, including the original labels and pseudo-labels, Classifier 1, Classifier 2 and Classifier 3 represent h_1, h_2 and h_3, Unlabeled Training Set represents D_U, Unlabeled Training Set 1, Unlabeled Training Set 2, Unlabeled Training Set t represent $D_{U_{t1}}$, D_{U_2} and D_{U_t}.Pseudo-labeled data 1, Pseudo-labeled data 2, Pseudo-labeled data l represents $\mathrm{d}_1^{\mathrm{U_t}}$, $\mathrm{d}_2^{\mathrm{U_t}}$ and $\mathrm{d}_\mathrm{l}^{\mathrm{U_t}}$;$\mathrm{C}$ represents the trained classifier after iteration T times.

Algorithm 1 Heterogeneous-training

```
Input: original labeled balanced training set $D_i$ unlabeled training set $D_U$ , validation set $D_V$ and test set $D_t$ , number of subsets of unlabeled training set $\mathrm{n}$, number of iterations $t_0$ , sample category proportion upper bound $P_h$ , lower bound $P_l$ , fine-tuning $\mathrm{step}$ .
Output: final classifier H, test result $F_1$ .
(1) Use $h_1$, $h_2$ and $h_3$ to preliminarily determine $D_U$ and get $L^0$.
(2) Calculate $Per$ in $L^0$ for m categories.
(3) Divide $D_U$ into $\mathrm{n}$ subsets at random, use $h_1$, $h_2$ and $h_3$ to determine $D_U$ , get $D_{U_t}$.
(4) For each patent text $\mathrm{d}_1^{\mathrm{U_t}}$ in $D_{U_t}$, $D_i += \left\{\mathrm{d}_1^{\mathrm{U_t}}\right\}$, $D_U -= \left\{\mathrm{d}_1^{\mathrm{U_t}}\right\}$ .
(5) If i (categories in $D_i) > P_h$ , $per_{h_k}^{c_i} += step$, else $per_{h_k}^{c_i} -= step$.
(6) Repeat steps (3) to (5); traverse through all the patent texts in $D_{U_t}$ .
(7) Clip $\mathrm{D_i}$ with RemoveOnly and calculate the performance.
(8) Retrain $h_1$, $h_2$ and $h_3$ with updated $\mathrm{D_i}$ ; validate $\mathrm{D_v}$ ,recording the result.
(9) Repeat steps (4) to (10) until the number of iterations is equal to $t_0$ .
(10) Take the best validation result among the 3 classifiers as the final classifier $\mathrm{H}$, test it on $D_t$ , and output $F_1$ .
```

AGNews[26] is a huge collection of news articles, divided into four coarse-grained topics, such as business, politics, sports and technology.

Books[27 28] is a data set containing book descriptions, user-book interactions and user book reviews collected from Goodreads 3, a popular online book review website.

The dataset statistics and the corresponding noise ratios for the initial pseudo-labels are provided in Table 1.

Table 1 Dataset statistics

Dataset	#Docs	#labels	#Noise Ratio(%)
NYT-Coarse	13081	5	11.51 †
NYT-Fine	13081	26	31.77
20News-Coarse	17871	5	12.48
20News-Fine	17,871	17	25. 69
AGNews	120,000	4	16. 26
Books	33,594	8	37. 35 †

4.3 | Compared Methods

We compare with several label selection methods mentioned below:

Tri-training [10], Tri-training with different classifiers, Tri-training+RemoveOnly [24], Tri-training with updated probability thresholds and Tri-training with "admission condition".

We consider the same number of samples in each iteration as Heterogeneous-training for all the above baselines because we cannot adjust individual thresholds for each dataset. There is no human annotated data in a weakly supervised setting and a fixed threshold does not work for all datasets due to the different prediction probability distributions in different datasets. Therefore, for controlled experiments and fair comparisons, we consider the same number of samples as Tri-training_ew in each iteration.

4.4 | Experimental Settings

The selection of hyperparameters in general text classification experiments is performed by manual experience or k-fold cross-validation. In this study, to avoid excessive complexity of the algorithm,some hyperparameters are determined by manual experience. We set the number of iterations t=10,the upper bound , the lower bound , and the fine-tuning . We focus on the effect of the choice of the number of subsets n of the unlabeled training set on the experimental results. If the number of subsets is set too large, the time complexity of the algorithm may be too high and the classification effect will not be improved significantly;if the number is set too small,it may not be accurate to discriminate the unknown samples due to insufficient iterations of the classifier,and it is measured that n is set to 10. For the classifier, we choose SVM, Naive Bayes and XGBoost to classify the text.

4.5 | Evaluation Metrics

The expressions for the accuracy and completeness are defined as:

$$\mathrm{P} = \frac{\mathrm{a}}{\mathrm{a}+\mathrm{c}} \times 100\% \tag{8}$$

$$\mathrm{R} = \frac{\mathrm{a}}{\mathrm{a}+\mathrm{b}} \times 100\% \tag{9}$$

In the expressions, a is the number of texts that actually belong to a category and are predicted by the classifier to be in that category; b is the number of texts that actually belong to a category and are predicted by the classifier to be in other categories; c is the number of texts that actually do not belong to a category but are predicted by the classifier to be in that category. The accuracy rate and the completeness rate reflect two different aspects of automatic patent classification, and generally speaking, the accuracy rate and the completeness rate are contradictory, i. e. , it is impossible to improve these two indicators at the same time, so they should be considered together. The evaluated value is used as the evaluation index of the experiment, and the expression is:

$$\mathrm{F1} = 2\mathrm{RP}/(\mathrm{R}+\mathrm{P}) \tag{10}$$

4.6 | Model Sensitivity

Figure 2 shows Heterogeneous-training method with different number of unlabeled training sets and different number of initial training sets. It aims to investigate the influence of the change of n on the classification effect under different number of initial training sets. Set n=6, 7, 8, 9, 10;

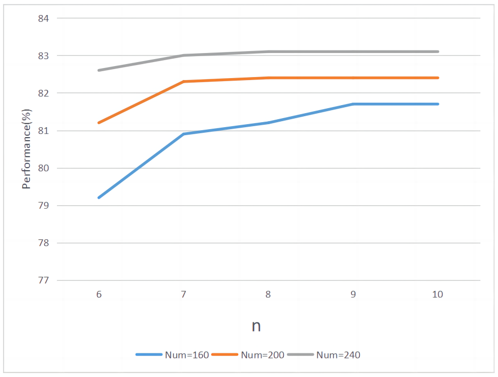

Figure 2 Heterogeneous-training with different number of unlabeled training sets and different number of initial training sets.

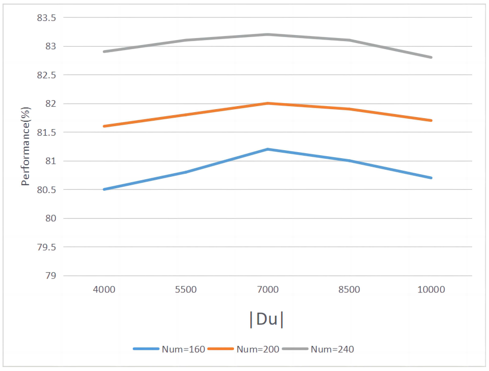

Figure 3 Heterogeneous-training of different numbers of unlabeled samples and different numbers of initial training sets.

Table 2 The results of the evaluation on six datasets.

	Coarse-grained Datasets				Fine-grained Datasets	
Method	NYT-Coarse	20News-Coarse	AGNews	Books	NYR-Fine	20News-Fine
Tri-training	78. 1	67. 4	63. 8	47. 8	70. 4	65. 3
Tri-training+different classifiers	79. 8	69. 2	63. 4	48. 6	72. 3	66. 8
Tri-training+RemoveOnly	84. 4	72. 6	66. 1	49. 7	71. 2	66. 2
Tri-training+updated probability thresholds	81. 6	70. 7	69. 7	47. 2	73. 5	67. 2
Tri-training+"admission condition"	80. 2	71. 9	66. 3	48. 1	72. 8	65. 8
Heterogeneous-training	**85. 3**	**76. 3**	**70. 9**	**52. 3**	**75. 6**	**69. 4**

Set the feature dimension to 450; 160, 200, and 240 text segments in D_1 constitute the original marked training set D_i, D_2 as the verification set D_v, D_3 as the test set D_t, and D_4 as the unlabeled training set D_U. The data sets are all provided from NYT-Coarse.

As can be seen from the figure, when there are relatively few labeled training samples, the Heterogeneous-training algorithm proposed in this study can use unlabeled patent samples to enhance the ability of supervised learning. With the number of three initial training sets, when n is from 6 to 7, $F1$ changes greatly; When n continues to increase, the improvement effect of $F1$ is not obvious. This shows that in order to obtain the best performance and lower time complexity at the same time, n should not be too large or too small. When the number of initial training sets continues to increase, $F1$ is improved, which shows that properly increasing the initial training sets can improve the classification effect of this algorithm.

Figure 3 compares the Heterogeneous-training of different numbers of unlabeled samples and different numbers of initial training sets to investigate the number of unlabeled samples under different numbers of initial training sets.The influence of the change on the classification effect. Set n=8, and set the feature dimension to 450; 160, 200 and 240 text segments in D_1 are used to form the original marked training set D_i, D_2 is used as verification set D_v, D_3 is used as test set D_t, and D_4 to D_8 are used as unlabeled training sets.The data sets are all provided from NYT-Coarse.

As can be seen from the figure, under the three initial training sets, when the number of unlabeled training sets is 7 000, $F1$ of Heterogeneous-training algorithm all reaches its peak value, which is 0.812,0.820, 0.832. Under different initial training sets, with the increase of unlabeled training sets, $F1$ shows a trend of first increasing and then decreasing, which indicates that the number of unlabeled training sets should not be too large or too small if the algorithm wants to get the best performance. Increasing the number of unlabeled patent samples in a certain range can improve the classification effect of the algorithm.

4.7 | Results

We present a summary of the results of the evaluation by using different classifiers and selecting methods in Table 2.Initial pseudo-labels are generated using String-Match[5]. Micro and Macro-F1 scores are used as evaluation metrics. Three random seeds were used for each experiment, and the average values and its corresponding standard deviations were given in percentage form. In order to make a fair comparison, we consider the same number of samples for each baseline as the Heterogeneous-training. Heterogeneous-training that outperforms the standard is rendered bold. All experiments are run on three random seeds, and mean, standard deviations are reported as percentages.

Experimental results show that the performance is significantly improved for all classifiers on fine-grained datasets. In some cases, such as NYT-Coarse, F1 improved by 8. 9. However,we also find that some methods are low in some cases. This may be due to the fact that the number of noisy labels is higher in fine-grained data sets, which leads to higher noise. It may also be that the selection of classification methods is not suitable for some data sets, and the classification effect of this algorithm on these data sets has not reached the expected performance compared with other classification algorithms.

5 | CONCLUSIONS

In this paper, we study and improve the Tri-training algorithm, and propose a new improved model Heterogeneous-training and apply it to the field of text classification. Heterogeneous-training can improve the accuracy of text classification without changing the training set. By dynamically updating the probability threshold and stricter conditions, the accuracy of this algorithm is greatly improved compared with that before improvement. Experimental results show that, compared with the original algorithm, Heterogeneous-training with three different classifiers is significantly better than Tri-training algorithm in the index F1 of experimental data set. However, the improved algorithm still has shortcomings. In the case of a small initial training set, the performance obtained with a smaller training set may be lower, so more mislabeled data may be generated in the classification process. How to eliminate noise better will be the focus of the future research on three training algorithms. Moreover, more superparameters in the algorithm are determined by human experience, and this problem should be solved in future research. Furthermore, we also expect the Heterogeneous-training algorithm to play a role in the field of deep learning.

ACKNOWLEDGMENTS

This work has been partially supported by Talent introduction project of Xihua University (Z202104), the Opening Project of Intelligent Policing Key Laboratory of Sichuan Province (Grant no. ZNJW2022KFMS004, ZNJW2022KFQN002, ZNJW2022ZZMS001, ZNJW2023KFQN007) and SCU-Luzhou Municipal People's Government Strategic Cooperation Project (Grant no. 2020CDLZ-10).

References

1. Agichtein E, Eskin E, Gravano L. Combining Strategies for Extracting Relations from Text Collections. In: ACM. ; 2000.
2. Li X, Roth D, Tu Y. [Association for Computational Linguistics the seventh conference - Edmonton, Canada (2003.05.31-..)] Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - - PhraseNet. 2003; 4: 87-94.
3. Tao F, Zhang C, Chen X, et al. Doc 2 Cube : Automated Document Allocation to Text Cube via Dimension-Aware Joint Embedding. In: IEEE. ; 2018.
4. Meng Y, Shen J, Zhang C, Han J. Weakly-Supervised Neural Text Classification. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* 2018.
5. Mekala D, Shang J. Contextualized Weak Supervision for Text Classification. In: Association for Computational Linguistics. Association for Computational Linguistics; 2020; Online: 323–333
6. Dietterich TG. Ensemble Methods in Machine Learning. In: Springer. ; 2000.
7. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: ACM. ; 1998.
8. Qian Z, Huailiang L. An Algorithm of Short Text Classification Based on Semi-supervised Learning. *Data Analysis and Knowledge Discovery* 2013; 29: 30-35.
9. Goldman SA, Zhou Y. Enhancing Supervised Learning with Unlabeled Data. In: International Conference on Machine Learning. ; 2000.

10. Zhou ZH, Li M. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 2005; 17: 1529-1541.

11. Saito K, Ushiku Y, Harada T. Asymmetric Tri-training for Unsupervised Domain Adaptation. In: International Conference on Machine Learning. ; 2017.

12. Miyato T, Dai AM, Goodfellow IJ. Adversarial Training Methods for Semi-Supervised Text Classification. *arXiv: Machine Learning* 2017.

13. Clark K, Luong MT, Manning CD, Le Q. Semi-Supervised Sequence Modeling with Cross-View Training. In: Empirical Methods in Natural Language. Association for Computational Linguistics; 2018; Brussels, Belgium: 1914–1925

14. Sachan DS, Zaheer M, Salakhutdinov R. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. In: AAAI. ; 2019.

15. Miyato T, Maeda iS, Koyama M, Ishii S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2019; 41: 1979-1993.

16. Xie Q, Dai Z, Hovy E, Luong T, Le Q. Unsupervised Data Augmentation for Consistency Training. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H., eds. *Advances in Neural Information Processing Systems*. 33. Neural Information Processing Systems. Curran Associates, Inc.; 2020: 6256–6268.

17. Chen J, Yang Z, Yang D. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. *ArXiv* 2020; abs/2004.12239.

18. Ganaie MA, Hu M, Tanveer M, Suganthan PN. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* 2022; 115: 105151.

19. Zhou Z. Machine Learning. *China Civil And Commercial* 2016; 03(No.21): 93-93.

20. Opitz D, Maclin R. Popular Ensemble Methods: An Empirical Study. *J. Artif. Int. Res.* 1999; 11(1): 169–198.

21. Dasgupta S, Littman ML, McAllester D. PAC Generalization Bounds for Co-Training. In: NIPS'01. Neural Information Processing Systems. MIT Press; 2001; Cambridge, MA, USA: 375–382.

22. Sun S. Local within-class accuracies for weighting individual outputs in multiple classifier systems. *Pattern Recognit. Lett.* 2010; 31: 119-124.

23. Wang S, Minku LL, Yao X. Resampling-Based Ensemble Methods for Online Class Imbalance Learning. *IEEE Transactions on Knowledge and Data Engineering* 2015; 27: 1356-1368.

24. Belkin M, Niyogi P, Sindhwani V. On manifold regularization. In: Artificial Intelligence and Statistics. ; 2005: 17-24.

25. Zhu X. Semi-Supervised Learning Literature Survey. In: Department of Computer Sciences, University of Wisconsin. ; 2005.

26. Zhang X, Zhao J, LeCun Y. Character-level Convolutional Networks for Text Classification. *arXiv: Learning* 2015.

27. Wan M, McAuley J. Item recommendation on monotonic behavior chains. *Proceedings of the 12th ACM Conference on Recommender Systems* 2018.

28. Wan M, Misra R, Nakashole N, McAuley J. Fine-Grained Spoiler Detection from Large-Scale Review Corpora. In: Association for Computational Linguistics. Association for Computational Linguistics; 2019; Florence, Italy: 2605–2610

29. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2016.