An Ensemble Learning-based model for Classification of Insincere Question

Zhongyuan Han[1], Jiaming Gao[2], Huilin Sun[1], Ruifeng Liu[1], Chengzhe Huang[1], Leilei Kong[1,*] and Haoliang Qi[1]

[1] Heilongjiang Institute of Technology Harbin, China
[2] Harbin Engineering University Harbin, China
{Hanzhongyuan,gaojiaming24,sunhuilin24,liuruifeng812,huangchengzhe9,Kongleilei1979,haoliang.qi}@gmail.com

Abstract. This paper describes the method for the Classification of Insincere Question(CIQ) in FIRE 2019. In this evaluation, we use an ensemble learning method to unite multiple classification models, including logistic regression model, support vector machine, Naive Bayes, decision tree, K-Nearest Neighbor, Random Forest. The result shows that our classification achieves the 67.32% accuracy rate(rank top 1) on the test dataset.

Keywords: Classification, Insincere Question, Ensemble Learning

1 Introduction

Community Question Answering(CQA) has seen a spectacular increase in popularity in the recent past. More and more people now use web forums to get answers to their questions. Given the scale of CQA forums on the web, identifying Insincere Questions becomes challenging by human moderators.

The FIRE 2019 CIQ task is a more detailed classification of the problems on the Quora platform. The question classification includes Rhetorical, Hate Speech, Hypothetical, Sexually Content, Other, Sincere questions. The evaluation data uses the Quora evaluation data provided by the website Kaggle, and the FIRE2019 organizer selects 898 training data, marked as six question categories. We use the TF-IDF of the terms as the text features, using logistic regression(LR), support vector machine(SVM), Naive Bayes(NB), decision tree(DT), K-Nearest Neighbor(KNN), Random Forest(RF) as the basic classification, and stacking-based ensemble learning method to train a learning algorithm to combine the predictions of these basic classification.

[*] corresponding author

2 Models

In this task, we use an ensemble learning method to solve the multiple classification task. Ensemble Learning is a way to combine multiple learning algorithms for better performance. The ensemble learning[1]includes two stages. Step 1: learn first-level classifiers. We select several base classifiers, such as LR(logistic regression), SVM(support vector, machine), NB(naive Bayes), DT(decision tree), KNN(k-nearest neighbor), RF(random forest), and train the independent classifiers. Step 2: learn a second-level meta-classifier. We use the output of the first-level classifiers as the new features. Next, use the new features to train the second level meta-classifier. The model structure is shown in **Fig. 1**.

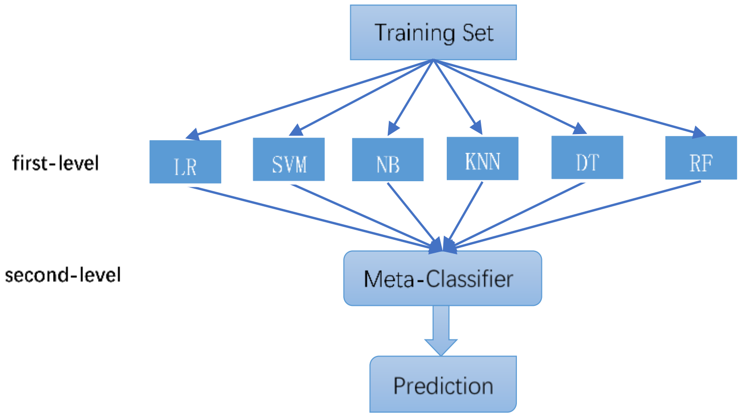

Fig. 1. Stacking-based Ensemble Learning

2.1 Logistics Regression

Logistic Regression[2] is a binary classification model. The input of the model is the K-dimensional eigenvector of the sample, the output of the model is the probability of a positive or negative class. For a given dataset T={(x_1, y_1),(x_2, y_2),$\cdots(x_n, y_n)$}, where $x_i \in R^n, y_i \in (0, 1)$, the model hypothetical function that the function is shown below:

$$h_\theta(x) = \mathrm{P}(y = 1|x; \theta) = \sigma(\theta^T x + b) = \frac{1}{1+exp(-\theta^T x+b)} \tag{1}$$

where θ represents the model parameter, ie the weight before each feature, b represents the bias, and σ represents the sigmoid function. In order to enable a multi-classification indicator the softmax function is needed to improve the logistic regression[3].

2.2 Support Vector Machine

The support vector machine[4] is a binary classification model. The basic model definition is the linear classifier with the largest interval in the feature space. For a given training dataset T={(x_1, y_1),(x_2, y_2),…(x_n, y_n)},$x_i \in R^n, y_i \in \{-1, +1\}$, the learning goal of SVM is to find a separate hyperplane in the feature space, divide the feature space into two parts. We use the linear separable support vector machine[5]. The classification decision function is:

$$f(x) = sign(w^* \cdot x + b^*) \tag{2}$$

where x represents the input feature, w^* represents the model weight, and b^* represents the bias. For the Multi-classification problem, the one-against-one[6] method can be used to construct a binary classification boundary between i-class and j-class data, train a binary SVM to solve the Multi-classification problem.

2.3 Naïve Bayes Classifier

Naïve Bayes[7] is a classification method based on Bayes theorem and the independent assumption of feature conditions. For a given set of training data, the joint probability distribution $P(X, Y)$ in the dataset is learned. Naive Bayes made a conditional independence hypothesis for the conditional probability distribution, specifically:

$$P(X = x|Y = c_k) = \prod_{j=1}^{n} P\left(X^{(j)} = x^{(j)}\middle|Y = c_k\right) \tag{3}$$

The joint probability distribution $P(X, Y)$ can be obtained. When using Naive Bayes for classification, for the input x , the posterior probability distribution P$(X = x|Y = c_k)$ is calculated by the classification model, and the class with the largest posterior probability is output as the category of x.

2.4 K-Nearest Neighbor

K neighbors[8] are a basic classification and regression method. Given a training dataset, for the new input sample, find the K samples closest to the sample in the training dataset. Most of the k samples belong to which class, and these samples are classified as this class. For a given training dataset T={(x_1, y_1),(x_2, y_2),…(x_n, y_n)}, where x_i is the feature vector of the sample, $y_i \in \{c_1, c_2, \cdots c_k\}$ is a sample category The category, i = 1, 2, ..., N; the sample feature vector is x; output y is the category to which the sample belongs:

$$\mathrm{y} = argmax_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j), i, j = 1,2, \cdots, K; \tag{4}$$

Where I is the indicator function, ie I is 1 when $y_i = c_j$, otherwise, I is 0.

2.5 Decision Tree

The decision tree model[9] is a tree structure classification model. A decision tree consists of nodes and directed edges. There are two types of nodes: internal nodes and leaf nodes. Internal nodes represent a feature or attribute, and leaf nodes represent a class. Using a decision tree classification model, start from the root node, test a feature of the sample, and assign the sample to its children according to the test results. The sample is tested and assigned recursively until the leaf node is reached. Finally, the sample is assigned to the class of the leaf node.

2.6 Random Forest

Random forests[10] use a random approach to synthesize many decision trees into a forest, and each decision tree votes to determine the final category of the test sample at the time of classification. First, the bootstrap method is used to generate m training sets, then, for each training set, a decision tree is constructed. When the nodes find features to split, a part of the features are randomly extracted from all features, then find the optimal solution by the extracted features, applied to the nodes, split, and finally achieve the effect of multi-classification.

3 Experiments

3.1 Datasets

The evaluation organizing provides an enhanced subset of Quora questions that contain the fine-grained category labels previously defined the insincere question. The evaluation dataset includes 898 training samples and 101 test data. Each sample contains qid as an identifier, question text show the content of the question, and the target indicates the category. The tag value is 0 to 5, representing 6 categories. The "0" tag indicates Sincere questions, the "1" tag indicates rhetorical questions, the "2" tags indicate hypothetical questions, the "3" tags indicate Hate speech questions, the "4" tags indicate sexually explicit questions, and the "5" tags indicate other. Each tag occupies the following **Table 1**.

Table 1. Each Tag Occupies.

Question Category	Ratio	Quantity	TAG
Rhetorical questions	0.55	488	1
Hate speech questions	0.24	216	3
Hypothetical questions	0.11	98	2
Sexually explicit questions	0.04	38	4
Other	0.04	38	5
Sincere questions	0.02	20	0
ALL	1	898	ALL

3.2 Experiments Setting

Our evaluation method consists of three parts: data processing, model training, and ensemble learning. In the data processing stage, we remove the stop words, remove punctuation, stemming. In the model training stage, the text is converted into a tf-idf vector using TfidVectorizer in scikit-learn [11], which is used as the training feature of logistic regression, support vector machine and other models. Get the best performance from the model by adjusting the hyperparameters. Finally, using the ensemble learning method, which can further improve the prediction accuracy.

During the data preprocessing stage, we removed the extra spaces in the text, used the stop vocabulary provided by Stanford[1] to remove the stop words, used the NLTK toolkit[2] for stemming operations, and removed the punctuation of the sentence. Next, we used 898 training data to directly train the ensemble learning model and 101 test data for testing.

In the model training stage, we select logistic regression as the meta-classifier to learn a second-level classifier. Before using ensemble learning, we need to set the hyperparameter of each classifier. We use train data and validation data to training each independent classifier, adjust the hyperparameters to achieve the best performance of the independent classifier on the validation set.

We use scikit-learn[3] to perform feature extraction and model training. Use the TfidfVectorizer tool provided by scikit-learn to convert the text data into TF-IDF feature vector, using the logistic regression, support vector machine, naive Bayes, K-nearest neighbor, decision trees, randomForest models provided by the scikit-learn toolkit for training. Ensemble learning uses the brew toolkit[4] for model fusion. Brew uses the output of each classifier as a new feature value, uses a logistic regression model to learn the weights of each classifier, then outputs the classification results. Brew uses the liblinear library for parameter solving, internally using the coordinate descent optimization combined with L2 regularization to iteratively optimize the loss function. Model parameter settings as shown in the following **Table 2**.

Table 2. Experimental Parameter.

Model	parameter
Logistic Regression(LR)	max_iter=10,penalty=l2,solver=liblinear,tol=1e-4
Support Vector Machine(SVM)	decison_function_shape=ovo,C=1,kernel=rbf
Naïve Bayes(NB)	alpha=0.01
K-Nearest Neighbors(KNN)	n_neighbors=10
Decision Tree(DT)	max_depth=3,min_samples_leaf=1,criterion=gini
Random Forest(RF)	n_estimators=10,max_depth=3,criterion=gini
Ensemble Learning(EL)	layer1=[LR,SVM,NB,KNN,DT,RF],layer2=[LR]

[1] https://github.com/stanfordnlp

[2] http://www.nltk.org/

[3] https://scikit-learn.org/

[4] https://pypi.org/project/brew/0.1.3/

3.3 Experiment Results

The final ranking of this evaluation task is shown in **Table 3.**

Table 3. Top 3 evaluation results of FIRE 2019

Team	accuracy_
Team HGC(our method)	0.6732
Team Neuro	0.6732
Team IRLab_IIT	0.6435
Team 4	0.6237

The experiment results of each model on test data in the following **Table 4**.

Table 4. Experimental Result.

Model	accuracy_
Logistic Regression(LR)	0.6435
Support Vector Machine(SVM)	0.6435
Naïve Bayes(NB)	0.6138
K-Nearest Neighbors(KNN)	0.6336
DecisionTree(DT)	0.6633
RandomForest(RF)	0.5940
EnsembleLearning(EL)	0.6732

It can be seen from the table that ensemble learning has achieved the best performance and the performance of the decision tree in the performance of a single classifier is the best. Different classifiers can learn different data features, and ensemble learning can integrate the features learned by each classification and the advantages of each classifier. In addition, through experiments, we found that the performance of the logistic regression and support vector machines is stable, and the classification performance is not obviously different. Decision trees and random forests are sensitive to train and test data. For the ensemble learning model, the performance is improved by combining the advantages of each classifier. It is more stable and does not cause performance degradation due to data replacement. And it will not have large performance fluctuations. But at the same time, we also see that the serious imbalance of data has a large impact on the individual classifier, which in turn affects the performance of the ensemble model. How to reduce the influential impact of data imbalance and generate more reasonable feature space is the direction of our next research.

4 Conclusions

In this evaluation, we used an ensemble learning approach that incorporates multiple text classification models to solve the Insincere Question Classification. Using TF-IDF as a model input feature, the scikit-learn toolkit was used to train logistic regression, support vector machines, naive Bayes and other models, and the models were merged using the brew toolkit. In the process of participating in this evaluation, we also tried Embedding as the feature input. Other machine learning models and deep learning models have also been tested, but the experimental results are not very satisfactory. Finally, TF-IDF features combined with ensemble learning were selected. The method is the final submission result.

Acknowledgments

This research was supported by the National Social Science Fund of China (No.18BYY125).

References

1. Smyth, Padhraic, and David Wolpert. "Linearly combining density estimators via stacking." Machine Learning 36.1-2 (1999): 59-83.
2. Cox, David R. "The regression analysis of binary sequences." Journal of the Royal Statistical Society: Series B (Methodological) 20.2 (1958): 215-232.
3. Jiang, Mingyang, et al. "Text classification based on deep belief network and softmax regression." Neural Computing and Applications 29.1 (2018): 61-70.
4. Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297.
5. Scholkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, 2001.
6. Hsu, Chih-Wei, and Chih-Jen Lin. "A comparison of methods for multiclass support vector machines." IEEE transactions on Neural Networks 13.2 (2002): 415-425.
7. Ng, Andrew Y., and Michael I. Jordan. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes." Advances in neural information processing systems. 2002.
8. Weinberger, Kilian Q., and Lawrence K. Saul. "Distance metric learning for large margin nearest neighbor classification." Journal of Machine Learning Research 10.Feb (2009): 207-244.
9. Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.
10. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.
11. Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.