Text Classification Based on Traditional Supervised and Unsupervised Machine Learning Methods

1 Introduction

Text and language are essential tools for exchanging information (Hijazi et al., 2021) in face-to-face and virtual communication on the Internet. In this era of the Internet explosion, tens of texts are on the Internet. It is very inefficient to rely on manual text recognition, so people hope that text recognition can be carried out through machine learning algorithms. Text classification is an integral part of it.

This research compares and studies the performance of various machine learning algorithms based on a given data set. The research question is listed below. There are three databases given to this project, including raw Tweets content (Eisenstein et al., 2010), a dataset containing only one thousand features after TFIDF feature selection, and a dataset containing 348 features after embedding calculation (Blodgett et al., 2016). Each dataset provides a training set, development set, test set, and unlabeled set. The training and development sets contain training classifier targets and demographic labels. The test set is used for the final evaluation, and one of the goals of this research is the prediction accuracy of the classifier in the test set.

Research question – Compared with the single model of supervised learning, whether the ensemble model of supervised learning, unsupervised learning model, and semi-supervised learning model can improve the performance of the text classifier.

2 Literature Review

2.1 Text Classification Algorithms

Traditional text classification methods have five categories. Firstly, Naïve Bayes methods include Bernoulli N.B., Gaussian N.B., and Multinomial N.B. When the number of features is large, the performance of such methods will decrease. Furthermore, the size of the dataset indeed influences the performance of text classification (Liu et al., 2019). Secondly, KNN-based methods are usually very time-consuming, especially while the size of the dataset is huge. Thirdly, the SVM methods are more suitable for dealing with high-dimensional problems. Fourthly decision tree-based methods are good at dealing with noise. Lastly, integration-based methods consist of Adaptive Boosting, XGBoost and Cat Boost (Li et al., 2020). In addition to traditional methods, there are many deep learning methods based on neural networks, which would have better performance than traditional methods.

2.2 Feature selection

Usually, there is a large size amount of features in the text classification process. For example, in the raw text provided in this study, the vectorized feature matrix size is (4000, 5354). However, the performance of text classification algorithms is inversely parabolic as the number of features increases, as shown in figure1 below (Hijazi et al., 2021), which means that we need to control the number of features used to train the model.

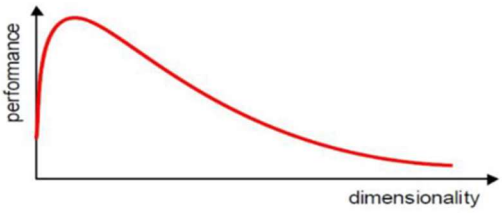

Figure 1 The relationship between classifier performance and dimensionality of training data (Hijazi et al., 2021)

There are two main feature selection methods: filter-based and wrapper-based (Hijazi et al., 2021; Thirumoorthy & Muneeswaran, 2022). Feature selection in the WebKB and BBC datasets improves text classification performance, and the accuracy increases more after feature selection in Naïve Bayesian classifiers (Thirumoorthy & Muneeswaran, 2022).

3 Method

Generally, as for traditional methods, text classification algorithms have four procedures: text preprocess, features extraction, classifier, and evaluation (Li et al., 2020). This research method is designed based on the above four procedures.

3.1 Preprocess and feature extractions

Machine learning can handle attribute data types which are nominal, ordinal, and numeric data. TFIDF dataset has been preprocessed for feature selection, containing 1000 highest TFIDF value words. As for the embedding one, its feature set finally consists of 384 dimensions through a pre-trained language learning model. Thus, only raw tweets need to be preprocessed in the given data sets (Blodgett et al. 2016), converting strings into an acceptable data type. The Count Vectorizer class under the feature_extraction module in the sklearn package (*API Reference — Scikit-Learn 1.0.2 Documentation*, n.d.) is chosen as the preprocess method. In doing so, the raw tweets reviews will be converted into a matrix of token counts.

Considering that the performance of some machine learning methods might be impacted by the large size of features, as Li mentioned in their paper (Danesh et al., n.d.). Consequently, as for raw tweet and TFIDF dataset, besides training models through all features, this study utilizes two feature selection methods to scale down the size of features before putting the data set into a machine learning. The two methods are SelectKBest and SelectFpr under *sklearn. feature_selection module*, selecting k highest-scored features referring to Chi-square test and choosing features that are significant at α level in false positive rate test, respectively. The wrong choice of parameters in the feature selection method might lead to poor model prediction performance. Therefore, a function is designed to find the parameters with the best performance in label prediction in set ranges. Afterwards, the feature selection methods under these optimal parameter settings will be used to finally predict the labels of the test set by comparing the performance of the dataset with full features and the dataset with filtered features in the same text classification algorithm to see if it is overfitting because too many features are used for training. After feature selection, a t-test was performed on the three datasets to test whether the two feature selection methods significantly improved the prediction accuracy.

3.2 Classifier

This research is mainly focused on comparing the text classification performance of various traditional machine learning single models, semi-supervised and unsupervised models, and embedded models on the given dataset. After a simple comparison and screening, three specific models from each category are chosen, which performed better or are mentioned in the literature for text classification (table1). Besides, the zero rule prediction is set as the baseline.

Category	Model
Baseline	0-Rule
Single Models	Logistic Regression
	Multinomial N.B.
	Bernoulli N.B.
Embedded Models	Adaptive Boost
	Gradient Boost
	Random Forest
Semi-supervised and Unsupervised Models	Self-learning
	Label Spreading
	K-means

Table 1 Display of chosen models in this research

3.3 Evaluation

Holdout, a popular strategy, is utilized as the evaluation strategy for this research and implemented by the train_test_split class in the sklearn package, with 80-20 split sizes and 0 random states. That means that, for all models, they train models and evaluate the model using the same set of instances, preventing distinctive model parameters and prediction performance due to different train and test instances.

Regards evaluation metrics, the supervised learning models are evaluated with the classification report, and the research mainly focuses on the accuracy score. As for unsupervised learning models, the purity of clusters is chosen as the evaluation metric.

4 Results

All selected machine learning models are trained for three given datasets, predicting sentiment for the test set, and the accuracy scores for each dataset and model are listed in Figure2 below. It is worth noting that the multinomial naive Bayes model cannot input negative values, but the value of the embedding dataset is not non-negative, so there is no result for this cell in the end. In general, single models have the highest prediction accuracy score, embedded models second, and semi-supervised the worst. Exceptionally, random forest performs the best in the embedding dataset, with a 0.675 accuracy score. Regarding the raw and TFIDF datasets, the supervised learning ensemble and semi-supervised models did not improve the prediction accuracy. In the TFIDF dataset, the baseline model achieves the highest accuracy score.

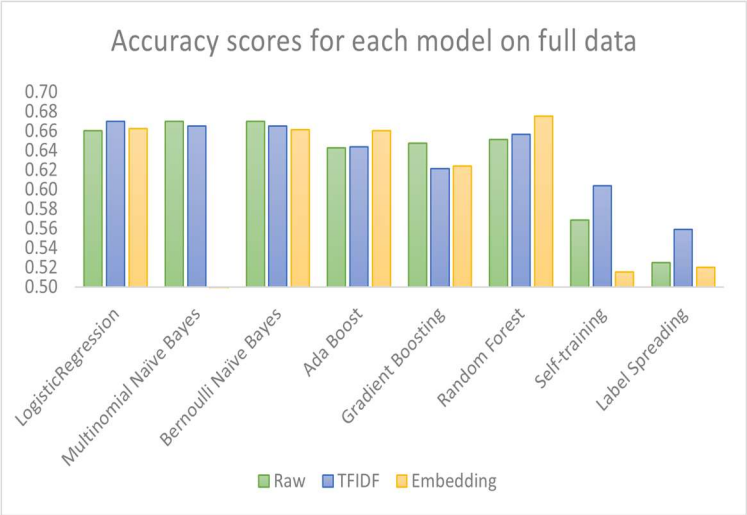

Figure 2 Accuracy scores on full feature set for all models

Table2 compares whether the performance of various models in the raw and TFIDF datasets has improved after feature selection. The preprocessed raw dataset attributes are in matrix format, while Label Spreading only accepts array-like input. Therefore, the Label Spreading model in the raw dataset is not considered. For both raw and TFIDF datasets, two feature selection methods slightly improve the performance of traditional single models and embedded models, but the prediction accuracy of semi-supervised models is improved. Table 3 indicates that at a 5% significant level, the two selection methods improved the accuracy score in the raw dataset. Although feature selection improved the performance of the models, none of the models performed significantly better. Still, traditional single models outperformed integration-based models and semi-supervised ones.

Data Set	Accuracy score	K-best	Fpr	Classifier
Raw	0.6600	0.6725	0.6713	Logistic Regression
Raw	0.6700	0.6700	0.6663	Multinomial N.B.
Raw	0.6700	0.6788	0.6788	Bernoulli N.B.
Raw	0.6425	0.6438	0.6425	Ada Boost
Raw	0.6475	0.6675	0.6613	Gradient Boosting
Raw	0.6513	0.6675	0.6613	Random Forest
Raw	0.5688	0.6488	0.6425	Self-training
Raw	0.5250	-	-	Label Spreading
TFIDF	0.6700	0.6713	0.6713	Logistic Regression
TFIDF	0.6650	0.6700	0.6700	Multinomial N.B.
TFIDF	0.6650	0.6700	0.6700	Bernoulli N.B.
TFIDF	0.6438	0.6438	0.6438	Ada Boost
TFIDF	0.6213	0.6538	0.6600	Gradient Boosting
TFIDF	0.6562	0.6638	0.6638	Random Forest
TFIDF	0.6038	0.6188	0.6163	Self-training
TFIDF	0.5588	0.6363	0.6350	Label Spreading

Table 2 Accuracy scores comparison pre and post feature selection

	Kbest		Fpr	
	t	**p-value**	**t**	**p-value**
Raw	-1.9060	0.0526	-1.6494	0.0751
Tfidf	-1.9411	0.0467	-1.9561	0.0457

Table 3 T-test results for accuracy scores improvement of two feature selection methods in two datasets

A good cluster means a high degree of purity and a low degree of entropy. However, as shown in Table 4 below, the K-means cluster does not perform well in all datasets. The raw dataset has the highest purity, while the TFIDF dataset has the lowest entropy.

Data Set	Purity	Entropy
Raw	0.6225	0.6230
TFIDF	0.6063	0.5637
Embedding	0.6100	0.6109

Table 4 Purity of k-means clustering model in three datasets

5 Discussion

The models selected for this research did

not perform well on all three datasets. First of all, the reason behind this may be that the given datasets are linearly inseparable, while the models chosen in this research are linear. The data provided is multidimensional, so there is no way to discern whether it is linearly separable by drawing a graph. Future research can determine whether the given datasets are linearly divisible by detecting whether the convex hull intersects. Secondly, variances of trained models might be high due to the massive size of training instances, around 4000 in the three given datasets. Thirdly, there may be outlier data in the training sets, which may mislead the classifier to train biased model parameters.

Moreover, the random forest performs the best among the selected models since the random forest is a nonparametric model. The random forest model can handle high-dimensional data and large training datasets well, and the risk of overfitting is reduced by averaging the decision trees. Besides, the embedded and unsupervised models did not perform better than traditional single machine learning models. Firstly, integration-based models do not always perform better than single models but reduce the risk of poor model selection. Secondly, the reason for the poor performance of the integrated models may be that the classifiers included in the integrated model mispredict the identical instances in the datasets.

Human language is complex and emotionally rich, and machines cannot determine the connotation and emotion of words or sentences based on the text itself, which is where the difficulty of natural language processing lies. This study's K-best and Fpr(false positive rate) methods select mechanical features based on the score function. The final selected subset of features may not help the machine determine the sentiment of the text and significantly improve the performance of the machine learning models. Furthermore, the feature selection method might be selected inappropriately, and perhaps casual feature selection (Shan et al., 2020) and TFDM (term frequency distribution measure) (Thirumoorthy & Muneeswaran, 2022) would have improved more classification performance in the dataset of this study.

As for K-means clustering, it is sensitive to noise points, which means that the predicted centre of each cluster would be far away from the actual centre due to outliers. Figure3 indicates that TFIDF and embedding datasets indeed contain some outliers. Besides, one assumption of k-means is that the data of each cluster follows a high-dimensional spherical distribution. In other words, when the datasets used for training and testing do not follow this distribution, k-means is more likely performing worse.

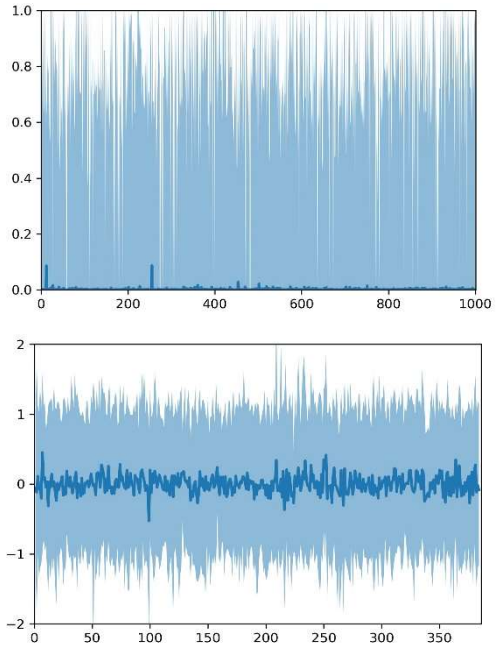

Figure 3 Minimum, maximum and mean of TFIDF and embedding datasets

Build a dictionary consisting of words that could represent the sentiment preference and assign specific weights to each word, using this dictionary to complete text extraction from raw text. Then traditional machine learning models' performances of text classification may be improved.

6 Conclusions

This paper mainly realizes the methods of traditional machine learning models, such as Naïve Bayes methods, integration-based models, and semi-supervised models, to deal with the classification of the tweet reviews text. It is not expected that integration-based models have higher accuracy than single models, especially are Naïve Bayes model classification predictions. After feature selection, the boosted dataset then various classifiers did not perform as well as expected.

7 References

API Reference — scikit-learn 1.0.2 documentation. (n.d.). Retrieved May 1, 2022, from https://scikit-learn.org/stable/modules/classes.html#

Blodgett, Su Lin, Green, Lisa, and O'Connor, Brendan. 2016. Demographic Dialectal Variation in Social Media: A Case Study of African-American English. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Danesh, A., Moshiri, B., & Fatemi, O. (n.d.). *Improve Text Classification Accuracy based on Classifier Fusion Methods*.

Hijazi, M. M., Zeki, A., & Ismail, A. (2021). Arabic text classification: A review study on feature selection methods. *2021 22nd International Arab Conference on Information Technology, ACIT 2021*. https://doi.org/10.1109/ACIT53391.2021.9677185

Li, Q., Yu, P. S., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., & He, L. 2021. (2020). A Survey on Text Classification: From Shallow to Deep Learning. *ACM Trans. Intell. Syst. Technol*, *37*(111), 39. https://doi.org/10.48550/arxiv.2008.00364

Liu, S., Tao, H., & Feng, S. (2019). Text Classification Research Based on Bert Model and Bayesian Network. *Proceedings - 2019 Chinese Automation Congress, CAC 2019*, 5842–5846. https://doi.org/10.1109/CAC48633.2019.8996183

Shan, G., Foulds, J., & Pan, S. (2020). *Causal Feature Selection with Dimension Reduction for Interpretable Text Classification*. https://doi.org/10.48550/arxiv.2010.04609

Thirumoorthy, K., & Muneeswaran, K. (2022). Feature Selection for Text Classification Using Machine Learning Approaches. *National Academy Science Letters*, *45*(1), 51–56. https://doi.org/10.1007/S40009-021-01043-0/TABLES/1