An Automated Essay Scoring model Based on Stacking Method

Chenchen Li[1]
School of Optoelectronic and Communication Engineering
Xiamen University of Technology
Xiamen, China
lcc@s.xmut.edu.cn

Lin Lin[1]
School of International Languages
Xiamen University of Technology
Xiamen, China
linlin@xmut.edu.cn

Wei Mao
School of Optoelectronic and Communication Engineering
Xiamen University of Technology
Xiamen, China
maov@s.xmut.edu.cn

Liu Xiong
School of Optoelectronic and Communication Engineering
Xiamen University of Technology
Xiamen, China
xiongliu@s.xmut.edu.cn

Yongping Lin(Corresondping Author)
School of Optoelectronic and Communication Engineering
Xiamen University of Technology
Xiamen, China
yplin@t.xmut.edu.cn

2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI) | 978-1-6654-8223-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/SEAI55746.2022.9832246

***Abstract*—In the latest two decades, thanks to AI technology, Automated Essay Scoring (AES) technology has also been rapidly developed. This technology to analyze and score essays automatically is a hot spot for the application of natural language processing in the field of education. Firstly, different encoding methods are used to obtain the lexical vectors of the text. Secondly, features of the seven aspects of the text are fully extracted. Then the comparative analysis of different ensemble learning models was studied on the English essay scoring competition dataset of Kaggle. Finally, a model based on the stacking method is proposed. The results show that the model based on the stacking method achieves the best results on all datasets. It improves the average QWK values on eight subsets by 1.0%∼2.8% compared to the baseline models of neural networks and feature-engineered.**

***Index Terms*—Automated Essay Scoring, Ensemble learning, Stacking**

I. INTRODUCTION

There are various ways to test student English proficiency. English essays play an important role. However, due to the subjective nature of manual scoring, reasonable assessment tends to require heavy labor and resources [1]. Therefore, fair and convenient Automated Essay Scoring (AES) technology is an important development direction in the field of education. With the development of artificial intelligence, natural language processing has been applied to all walks of life. AES technology is a hot application of natural language processing in the field of education [2]–[5].

Early AES technology includes Project Essay Grade (PEG) [6] and Intelligent Essay Assessor (IEA) [7], which were based on surface features and implicit semantic analysis, respectively. Electronic Essay Rater (E-rater) technology has the advantages of the first two [8]. AES technology has been applied in some scoring platforms such as batching.com.

The most advanced AES technology today has two implementation methods. One is feature-engineered method [9], [10]. The other is based on neural network method [1], [11]–[15]. The feature-engineered method first needs to extract the surface features of the text, then concatenate them with the vector representation of the text. Finally, it predict the results by different regression algorithms. The typical method is the Two-Stage Learning Framework (TSLF) proposed by Liu et al [10]. This method does not need a large dataset and the prediction results have strong interpretability. However, this method neither extracts the deep features of the text well nor has good generalization ability. The neural network method predicts the vector representation of the text directly, which can analyze the deep features of the text and has good generalization ability. However, the prediction results do not have good interpretation and need a large dataset for prediction. Among the typical methods are Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), LSTM+CNN [16], LSTM+CNN+Attention [17], etc.

In this paper, the models based on feature-engineered method and neural network method on the public Automated Student Assessment Prize dataset (ASAP) were compared and analyzed. The stacking method is a method that can fuse

[1] Authors contributed equally.

978-1-6654-8223-3/22/$31.00 ©2022 IEEE

multiple machine learning models. Using the Stacking method can combine the advantages of various models to achieve better results than individual model. So a model based on the Stacking method was proposed in this paper. The results show that the model based on Stacking method outperformed the other models. This study provides some references for the development of AES technology in the future.

II. Related Methods

This study compares and analyzes the prediction effectiveness of models based on different ensemble learning methods while using the Quadratic Weighted Kappa (QWK) value to evaluate the performance of the models.

A. *Ensemble learning*

Ensemble learning, as the name implies, is to achieve performance improvement by integrating multiple weak learners into strong learners. In this paper, three main methods are used to implement ensemble learning method, namely Bagging, Boosting, and Stacking.

Among the Bagging methods, Random Forest (RF) is a widely used algorithm, which samples the samples randomly and has better overfitting prevention and lower variance [18]. Gradient Boosting Decision Tree (GBDT) is one of the Boosting methods, which adds up the results of multiple regression trees as the final answer and has a strong generalization ability [19]. Based on this, Tianqi Chen et al. proposed the eXtreme Gradient Boosting model (XGBoost), which greatly improves the speed and efficiency [20]. The biggest difference between XGBoost and GBDT is the definition of the objective function. The XGBoost objective function is defined in the equation 1. Where l is the loss function (for example, the square loss function: $l\left(y_i, y^i\right) = [(y_i - y^i)]^2)$), $\Omega\left(f_t\right)$ is the regularization (including L1 regularization and L2 regularization), C is a constant.

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right) + C \qquad (1)$$

The Stacking method is implemented by secondary learning of the prediction results of the base learner by the meta-learner, which improves the prediction effect greatly although it increases the complexity of operation to some extent. In this paper, Random Forest, GBDT, and XGBoost algorithms are used for the base learner and the Ridge algorithm is used for the meta-learner. Before training, the dataset is first divided into the training set and the test set. The base learner first gets the prediction results of each part of the training set after 5-Fold cross-validation, and these results are concatenated together as the training set of the meta-learner. At the same time, the prediction results of the test set are obtained by the base learner, and they are averaged as the test set of the meta-learner. Finally, the final prediction results are obtained by the meta-learner.

B. *Kappa scoring metrics*

Quadratic Weighted Kappa (QWK) is a common measure of AES technology strengths and weaknesses, which is calculated based on a confusion matrix of predicted outcomes and true values. The weight is calculated as follows:

$$\omega_{i,j} = \frac{(i-j)^2}{(N-1)^2} \qquad (2)$$

The kappa coefficient calculation formula is as follows:

$$k = 1 - \frac{\sum_{i,j} \omega_{i,j} O_{i,j}}{\sum_{i,j} \omega_{i,j} E_{i,j}} \qquad (3)$$

Among them, $O_{i,j}$ represents the number of times that the true value i is predicted to be j. $E_(i,j)$ represents the product of the probabilities that the true value is i and the predicted value is j.

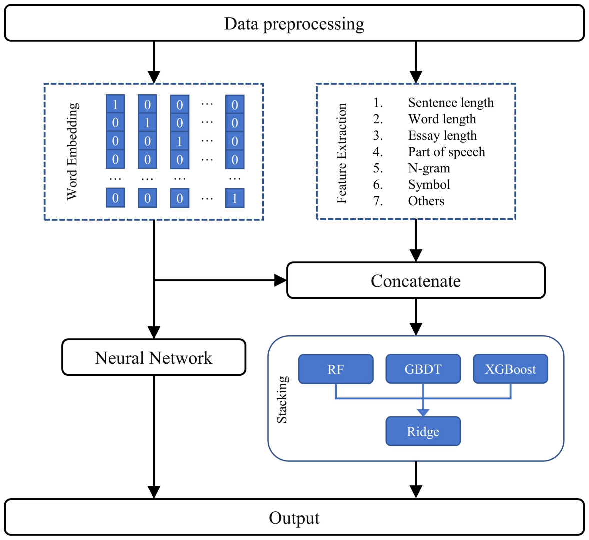


Fig. 1: The general structure of the model.

III. Model

A. *Dataset*

The public dataset of the Kaggle ASAP contest, which is widely used in the field of automated essay scoring, was used in this paper. The essays included in ASAP were written by students in grades 7-10. The dataset is divided into eight subsets based on the content of the essays. Each subset contains an essay prompt file and contains multiple essays on related topics. Details of the datasets are shown in Table II.

B. *Model framework*

The overall design of the model is shown in Fig. 1. The neural network method consists of three parts. The experimental data is preprocessed to get the input for the word embedding module. The word embedding module can be implemented by

TABLE I: Handcrafted features.

Feature Type	Examples
Sentence length	The average number of words per sentence, sentences contain more than 30 words, sentences contain more than 20 words, and sentences contain less than 10 words.
Word length	The average number of characters per word, words contain more than 7 characters, and words contain more than 4 characters.
Essay length	Total characters, total words, total sentences.
Part of speech	Numbers of nouns, verbs, adjectives, and adverbs.
N-gram	Numbers of unigrams, bigrams, trigrams.
Symbol	Numbers of exclamation marks, question marks, and commas.
Others	Numbers of distinct words, numbers of stop words, numbers of misspelled words, and the total number of lemmatized words.

TABLE II: ASAP dataset

essay_set	essay_count	Average length	Score range
D1	1783	350	2∼12
D2	1800	350	1∼6
D3	1726	150	0∼3
D4	1772	150	0∼3
D5	1805	150	0∼4
D6	1800	150	0∼4
D7	1569	250	0∼30
D8	723	650	0∼60

the Lookup Table method. The output of the word embedding module goes through the neural network model to get the final prediction results. The feature-engineered method is composed of four parts: data preprocessing module, word embedding module, feature extraction module and Stacking module. The experimental data are first passed through the data pre-processing module to remove the influence of data anomalies on the prediction results. The word embedding module and the feature extraction module are used to obtain the vector representation of the essay and essay-related features, respectively. The outputs of both are concatenated together and used as the input of the Stacking module. Finally, the final prediction results are learned by the Stacking module.

1) Data preprocessing: In the field of natural language processing, data pre-processing is one of the first and most important parts. A scientific approach to data pre-processing is a prerequisite for achieving a good model. Data preprocessing consists of two steps. The first step is clean up missing data. There may be some missing data in the essay data, which is detrimental to the training of the model. The missing data items are removed in this paper. The second step is uniform case. The machine learning process may treat words with different cases as two different words, which may lead to errors in prediction results. Therefore, in this paper, we reduce the error in this area by standardizing the case.

2) Word embedding: The word embedding module generates text vectors corresponding to essays. The feature-engineered method uses One-hot encoding to generate word vectors. The neural network method generally uses Lookup Table or Word2Vec to generate word vectors. Since neural networks can learn deep text features in articles without extracting features, features can be extracted and output scores can be predicted directly by the LSTM model.

3) Feature extraction: As shown in Table I, the extraction of text features mainly includes seven aspects: Sentence length, Word length, Essay length, Part of speech, N-gram, Symbol, and others. Machine learning algorithms that combine the above features can fully take into account the richness and complexity of the vocabulary and sentences in essays. So machine learning models with feature-engineered can achieve better performance. The feature is then concatenated with the feature vector generated by the word embedding module as the input for the next step.

4) Stacking: The text vector after feature-engineered can be trained by ridge regression, random forest, GBDT, XGBoost, etc. to get the predicted scores. In this paper, the Stacking method is used to get the final predicted scores. The stacking method is implemented by stacking multiple strong learners on top of weak learners. In the Stacking method, the strong learner is used as the base learner to learn the initial prediction of the text, and the weak learner is used as the meta-learner to learn the final prediction of the text. In this paper, Random Forest, GBDT, and XGBoost are used as base learners and Ridge regression is used as the meta-learner.

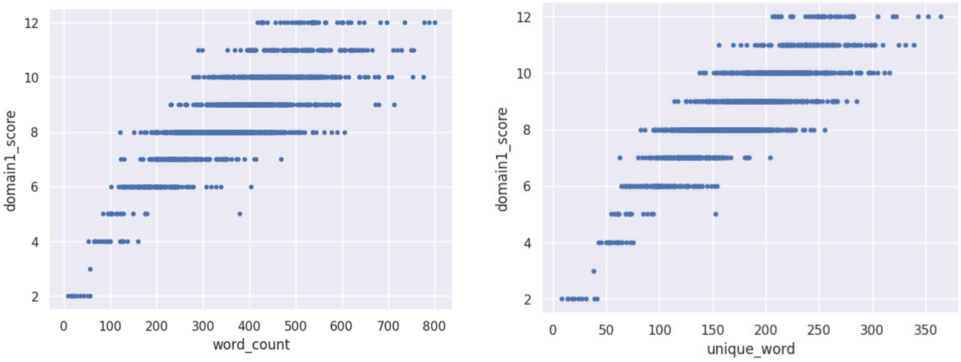
C. Parameter configuration

The feature-engineered method uses the CountVectorizer() function to convert the set of text documents into a token count matrix to generate the One-Hot vector corresponding to the text. In this paper, only the 10000 most frequent words in the thesaurus are selected, so $max\_features$ is set to 10000. At the same time, 1 to 3 words are selected as the combination method to form word frequency labels, so $ngram\_range$ is set to $(1, 3)$. The model first uses linear regression and ridge regression with regularization for simple prediction, and $1/5$ of the total number of samples is randomly selected as the test set, and the rest is used as the training set to predict the score results and calculate the QWK value of the model.

IV. Results and Discussions

Take the D1 subset as an example, the model obtained features of each essay such as char_count and adv_count in Table III after feature extraction. Fig. 2 shows the relationship between unique words count and the total score of the essay and the relationship between word count and the total score of the essay. From the figure, it can be seen that using more different words can get higher scores. This phenomenon coincides with reality and also provides some ideas for students to improve their English essay scores.

Again, take the D1 subset as an example. Fig. 3 shows the heat map of the confusion matrix of predicted and true values. The model based on the Stacking method can be visualized

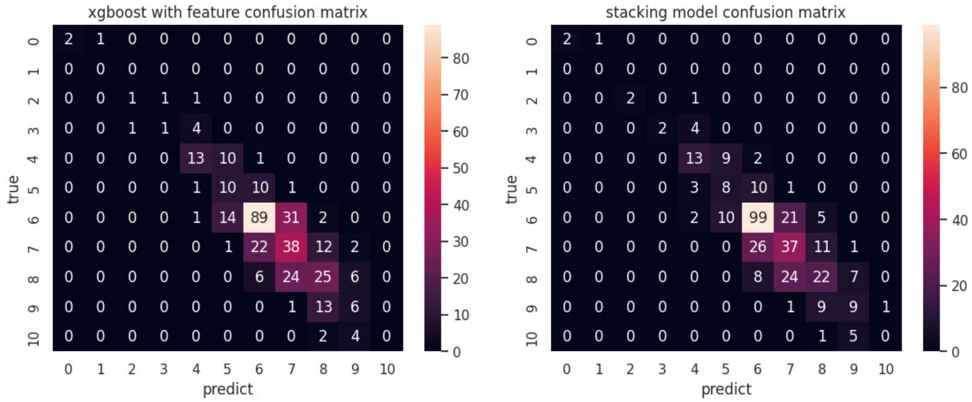


(a) Word count. (b) Unique word.

Fig. 2: Relationship between text features and essay scores.

TABLE III: Feature extraction results for the first five records on D1 subset.

essay_id	essay_set	essay	final_score	char_count	word_count	. . .	verb_count	adv_count
1	1	Dear local newspaper, I think effects computer...	6	1441	344	. . .	18	24
2	1	Dear I believe that using computers will benef...	7	1765	413	. . .	20	19
3	1	Dear, More and more people use computers, but ...	5	1185	276	. . .	20	16
4	1	Dear Local Newspaper, I have found that many e...	8	2284	490	. . .	39	29
5	1	Dear I know having computers has a positive ef...	6	2023	469	. . .	32	36



(a) XGBoost. (b) Stacking.

Fig. 3: Heat map of the confusion matrix of predicted and true values of the ASAP dataset D1.

and it predicts higher essay scores as true values compared to the model based on XGBoost method, which reflects the advantage of the Stacking method over the XGBoost method.

Table IV shows the comparison of the prediction results of different models on each subset of the ASAP dataset. In this paper, neural network methods such as LSTM, CNN+LSTM, CNN+LSTM+Attention, and feature-engineered methods such as Linear regression, Ridge Regression, and Random Forest were studied. The CNN+LSTM and CNN+LSTM+Attention models uses CNN to extract the local semantic features of the essay. Then the LSTM captures the deeper semantic features to achieve better results. CNN+LSTM+Attention is an attention mechanism added after the CNN layer and LSTM layer of CNN+LSTM so that higher weights can be assigned to key information. Overall, the highest average QWK values were achieved by the model based on the Stacking method.

The average QWK value of ridge regression is greatly improved compared with linear regression due to the addition of regularization. The simple neural network model is still slightly deficient compared with the feature-engineered model, while the multilayer neural network model captures deeper semantic features and achieves comparable results with the feature-engineered method. The Stacking-based model achieves the highest average QWK value, which illustrates the effectiveness of the Stacking method for improving model performance.

Table V shows the training time of each model on different subsets. The model based on the Stacking method achieves the best results but takes longer to train. This shows that the Stacking method increases the complexity of the model.

V. Conclusions

In this paper, the features of seven aspects of the essay were extracted. Meanwhile, One-Hot coding method and the

TABLE IV: Comparison of QWK scores on the ASAP dataset with the model based on the Stacking method and the baselines models.

Model	QWK								Avg QWK
	D1	D2	D3	D4	D5	D6	D7	D8	
LSTM	0.775	0.687	0.683	0.795	**0.818**	0.813	0.805	0.594	0.746
CNN+LSTM	0.821	0.688	**0.694**	0.805	0.807	**0.819**	**0.808**	0.644	0.761
CNN+LSTM+Attention	0.822	0.682	0.672	**0.814**	0.803	0.811	0.801	0.705	0.764
Linear	0.785	0.590	0.514	0.690	0.726	0.776	0.678	0.707	0.683
Ridge	0.802	0.628	0.579	0.793	0.751	0.786	0.731	0.721	0.724
RF	0.851	0.718	0.686	0.769	0.779	0.797	0.782	0.721	0.763
GBDT	0.852	0.711	0.676	0.790	0.786	0.796	0.778	0.693	0.760
XGBoost	0.859	0.693	0.652	0.797	0.787	0.811	0.784	0.724	0.763
Stacking	**0.863**	**0.719**	0.690	0.791	0.804	0.805	0.782	**0.737**	**0.774**

TABLE V: Training time of different models on each subset.

Model	Training Time (TT)								Avg TT
	D1	D2	D3	D4	D5	D6	D7	D8	
RF	45s	43s	30s	27s	28s	31s	25s	13s	30s
GBDT	47s	40s	37s	39s	38s	39s	34s	13s	36s
XGBoost	7s	5s	6s	6s	5s	5s	4s	2s	5s
Stacking	96s	103s	74s	77s	75s	78s	65s	28s	75s

Lookup Table method were employed to obtain the text representations based on the feature-engineered and the neural network model, respectively. The prediction results of different neural network methods and feature-engineered methods were studied. The experiments showed that the regularization process was effective in improving the model performance. Moreover, the multi-layer neural network model achieved better results than the single-layer model. The model based on the Stacking method achieved best results comparing the other models by fusing multiple regression methods. The experimental results showed that the average QWK value of the Stacking-based model was improved by 1.0%∼2.8% compared to the other baseline models. But this will increase the complexity of the model. Our method could be a useful tool for the future development of AES technology.

Acknowledgment

This work was supported by Natural Science Foundation of Fujian Province (2021J011216); Key Natural Science Foundation of Fujian Province (2020J02048) and Social Science Foundation of Fujian Province (FJ2021B161).

References

[1] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 1560–1569.

[2] Z. Xianbing, F. Xiaochao, R. Ge, and Y. Yong, "Automated english essay scoring method based on multi-level semantic features," *Journal of Computer Applications*, vol. 41, no. 8, p. 2205, 2021.

[3] J. Shin and M. J. Gierl, "More efficient processes for creating automated essay scoring frameworks: A demonstration of two algorithms," *Language Testing*, vol. 38, no. 2, pp. 247–272, 2021.

[4] J. C. Machicao, "Higher education challenge characterization to implement automated essay scoring model for universities with a current traditional learning evaluation system," in *International conference on information technology & systems*. Springer, 2019, pp. 835–844.

[5] M. Beseiso, O. A. Alzubi, and H. Rashaideh, "A novel automated essay scoring approach for reliable higher educational assessments," *Journal of Computing in Higher Education*, vol. 33, no. 3, pp. 727–746, 2021.

[6] M. D. Shermis and J. C. Burstein, *Automated essay scoring: A cross-disciplinary perspective*. Routledge, 2003.

[7] T. K. Landauer, "Automatic essay assessment," *Assessment in education: Principles, policy & practice*, vol. 10, no. 3, pp. 295–308, 2003.

[8] J. Burstein, J. Tetreault, and N. Madnani, "The e-rater automated essay scoring system," *Handbook of automated essay evaluation: Current applications and new directions*, pp. 55–67, 2013.

[9] W. Zhu and Y. Sun, "Automated essay scoring system using multi-model machine learning," in *CS & IT Conference Proceedings*, vol. 10, no. 12. CS & IT Conference Proceedings, 2020.

[10] J. Liu, Y. Xu, and Y. Zhu, "Automated essay scoring based on two-stage learning," *arXiv preprint arXiv:1901.07744*, 2019.

[11] C. Jin, B. He, K. Hui, and L. Sun, "Tdnn: a two-stage deep neural network for prompt-independent automated essay scoring," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1088–1097.

[12] J. Xue, X. Tang, and L. Zheng, "A hierarchical bert-based transfer learning approach for multi-dimensional essay scoring," *IEEE Access*, vol. 9, pp. 125 403–125 415, 2021.

[13] M. Beseiso and S. Alzahrani, "An empirical analysis of bert embedding for automated essay scoring," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 204–210, 2020.

[14] L. Xia, J. Liu, and Z. Zhang, "Automatic essay scoring model based on two-layer bi-directional long-short term memory network," in *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, 2019, pp. 133–137.

[15] S. Mathias and P. Bhattacharyya, "Can neural networks automatically score essay traits?" in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 85–91.

[16] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1882–1891.

[17] F. Dong, Y. Zhang, and J. Yang, "Attention-based recurrent convolutional neural network for automatic essay scoring." in *CoNLL*, 2017, pp. 153–162.

[18] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.