An Empirical Analysis of BERT Embedding for Automated Essay Scoring

Majdi Beseiso[1]
Department of Computer Science
Al-Balqa Applied University, Alsalt, Jordan

Saleh Alzahrani[2]
Information Systems Department, Al-Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

***Abstract*—Automated Essay Scoring (AES) is one of the most challenging problems in Natural Language Processing (NLP). The significant challenges include the length of the essay, the presence of spelling mistakes affecting the quality of the essay and representing essay in terms of relevant features for the efficient scoring of essays. In this work, we present a comparative empirical analysis of Automatic Essay Scoring (AES) models based on combinations of various feature sets. We use 30-manually extracted features, 300-word2vec representation, and 768-word embedding features using BERT model and forms different combinations for evaluating the performance of AES models. We formulate an automated essay scoring problem as a rescaled regression problem and quantized classification problem. We analyzed the performance of AES models for different combinations. We compared them against the existing ensemble approaches in terms of Kappa Statistics and Accuracy for rescaled regression problem and quantized classification problem respectively. A combination of 30-manually extracted features, 300-word2vec representation, and 768-word embedding features using BERT model results up to 77.2 ± 1.7 of Kappa statistics for rescaled regression problem and 75.2 ± 1.0 of accuracy value for Quantized Classification problem using a benchmark dataset consisting of about 12,000 essays divided into eight groups. The reporting results provide directions to the researchers in the field to use manually extracted features along with deep encoded features for developing a more reliable AES model.**

***Keywords*—Automated Essay Scoring (AES); BERT; deep learning; neural network; language model**

I. Introduction

Automated Essay Scoring (AES) involves the use of statistical models for extracting useful features from the essay and assigning grades in the numeric range. It helps to reduce human efforts in manual grading of essays and improve the effectiveness and efficiency of writing assessment. Several models have been proposed for automatic essay scoring in the recent past. Broadly, these models can be further categorized into two classes [1]. The first type of AES models belongs to feature engineering-based models. These models use manually extracted features from an essay in term of number of words, number of grammatical errors, number of unique vocabulary words, term frequency, inverse document frequency, etc. [2-3]. Feature engineering-based models have the benefits of using manually extracted features that can be easily explained and modified to adapt different scoring criteria. However, these model suffer from the limitation of lack of understanding some cement features leading to low accuracy of the models.

The second type of AES models is called an end to end models. These models are developed using machine learning or deep learning techniques [4, 5] based on some word embedding methods [6, 7]. The word embedding methods represent essay into low dimensional vectors. A dense layer follows the low dimensional vectors for transforming them into a deep encoded vector for further scoring of the essay. End to end models exhibits good performance for extracting semantic features and address the limitation of feature engineering models. However, these models are unable to integrate manually extracted features.

AES engine assigns a score to an essay based upon extracted features from the raw data of essays. The scoring process involves two phases [8]. The first phase consists of collecting the data for scoring by AES engine. The engine is trained based on some holistic rubrics that specify the satisfaction criteria of the essay. The rubrics consider different factors like grammatical errors, spelling mistakes, clarity, organization of the text, and Cohesion of the essay [9]. Kaggle competition has made AES data set available to the public. The second phase involves dividing the essay dataset into two data subsets for training and testing purposes. The training data set is a labelled data set used for developing a trained model of AES engine based upon the selected features of essay dataset. The trained model is further applied to the test data set for assigning them the labels as a score of the essay.

In this paper, we focus on manually extracted features as well as word embedding features of BERT model for analyzing the performance of but language model in automated scoring of essay. We conduct a set of experiments using word-embedding models along with the manually extracted features and compare their performances for automated scoring of essay using a benchmark dataset. The performance of different models is compared in terms of Kappa statistics and accuracy by considering the automatic scoring process as rescaled regression and quantized classification problem, respectively.

Rest of the paper is structured as follows. Section 2 highlights the background of AES and describes the different models developed for efficient AES. Section 3 describes the details of experiments, such as experimental setup, benchmark dataset and performance metrics. It provides comprehensive experimental mythology being following in this work. Section

4 presents results, analyses and compares the results with the existing approaches. Section 5 concludes this paper at the end.

II. Background

AES is considered as one of the most challenging problems in natural language processing (NLP). The significant challenges are the length of the essay, the presence of spelling mistakes affecting the quality of the essay. Several research efforts have been invested in the recent past for automated essay scoring [8, 10]. Initially, these research efforts involve the use of statistical methods based upon bag of words (BOW), use of Logistic regression method, and other probability-based methods. Some researches applied neural networks for automated scoring of essays using the word embedding method [6]. Embedding methods mainly work on characters words or sentences and transform them into n-dimensional vectors by preserving semantic features. It results in a conversion of character data into a sequence of n-dimensional data. The n-dimensional vector can be further used to create the model of different neural networks like LSTM, CNN and GRU [11]. These neural networks are the nonlinear models that are used to score the given essay based upon some scoring rubrics.

Ke et al. (2019) [12], Chen et al. (2010) [13], and Wang et al. (2018) [14] have summarized supervised and unsupervised learning-based embedding methods. In supervised learning methods for automatic scoring of essays, the researches considered AES problem as a regression problem and classification problem. Regression problem involves predicting the score of essay in the given numeric range. Classification problem involves the classification of essay to one of the predefined classes like medium, low and high. In case of regression, researchers use linear regression [15, 16], support vector regression [17, 18], and sequential minimal optimization (SMO, a variant of support vector machines) [19] for automatic scoring of the essays based upon different features. In the case of classification, researchers employed SMO [19], logistic regression [20] and Bayesian network classification [21] for classifying essays to their predefined classes. Many researchers also used neural networks for automated scoring of essays. Taghipour and Ng [22] proposed the first approach based on neural network for scoring essays. They used a series of words as input to convolutional layer and extracted n-gram features from essays. The extracted features represent local text dependencies among words. The extracted features are passed to LSTM layers for capturing long-term dependencies in the words of essay. Further, they concatenated vectors at different time intervals for feeding to a dense layer. Finally, they predicted the score of the essay after training of the model.

The above-cited research work uses different types of features like implicit features or explicit about scoring the essay automatically using different models. The performance of the model is mainly dependent upon the extent to which the extracted feature represents the given essay. Some researchers focused on manually extracted features, word2vec feature representation or embedding representation. In this work, we believe and hypothesize that both manually extracted features and deep-encoded features can contribute to enhancing the performance of AES models. Therefore, we conducted a comprehensive set of experiments in this work to evaluate word embedding in combination with manually extracted features and word2vec features.

III. Experiments

This section describes a comprehensive set of experiments conducted in this work to evaluate the performance of word embedding in combination with manually extracted features and word2vec features. It presents for experimental methodology by explaining different proposed in this work. Benchmark data set is used for comparing the performance of different models based upon different feature sets. This section also defines the set of performance metrics used to measure the performance of different models in this work.

A. Experimental Methodology

To conduct a comprehensive set of experiments, we followed the experimental methodology presented in Fig. 1. The proposed methodology consists of four modules, namely, essay raw data collection, feature extraction, scoring engine, and performance evaluation.

Raw data collection module collects the raw data of essays from the database and feeds into the feature extraction module. The feature extraction module can employ different types of methods for extracting relevant features that preserve the semantics of the essay. The features can be extracted manually, word2vec representation or by using the word embedding method. In this work, we focus on measuring the performance of AES models based upon different combinations of manually extracted features, word2vec representations, and word embedding using BERT model. We use 30 manually extracted features, 300-dimensional word2vec representation, and 768-word embedding features using BERT model and forms different combinations for evaluating the performance of AES models. Table I. summarizes manually extracted used in this work.

The different combinations of manually extracted features, word2vec representation and word embedding features are provided as input to AES engine for scoring the test essay data set after training of AES model based on training essay data set. The performance evaluation module analyses the performance of AES models based upon different combinations of manually extracted features, word2vec representation and word embedding features as presented in Fig. 1. Here, we fine-tuned the BERT model using different hyper-parameters. The optimal values used in this set of experiments are presented in Table II.

B. Benchmark Dataset

Most AES related research work used the Automated Student Assessment Prize (ASAP) dataset for evaluating AES models [1, 2, 3,]. This data set contains about 12,000 essays divided into eight groups. Essays in the data set are not assigned in the normalized score range. We assign scores that range from [2 - 12] to [10 - 60]. Essays in the data set also have a variable length ranging from 120 tokens to 500 tokens. Sentences have a length from 120 to 500 tokens. Each group of the data set contains about 700 to 1800 items.

To use the available data as benchmark dataset in our experiments, we normalize the essays score in the range of 0 to 10 by applying independent transformation for essay group. The resultant distribution of the scores in the data set is not uniform but seems like the normal distribution. Since the current work involved the comparison of the performance of the AES model in scoring rescaled regression problem and Quantized Classification problem, so we distributed dataset into three subgroups by approximating two quartile cut points. Each subgroup is replaced with its number in ascending order for obtaining a discrete score of 0, 1, or 2 effectively. We use 3-quantile subgroups discretization to produce far from equally populated subgroups due to skewed score frequencies in our experiments. A complete dataset has the frequencies per 3-subgroups in classification problem as presented in Table III.

In this work, we use this dataset as a benchmark dataset for evaluating the performance of different models based upon different combinations of manually extracted features, word2vec representation and word embedding features.

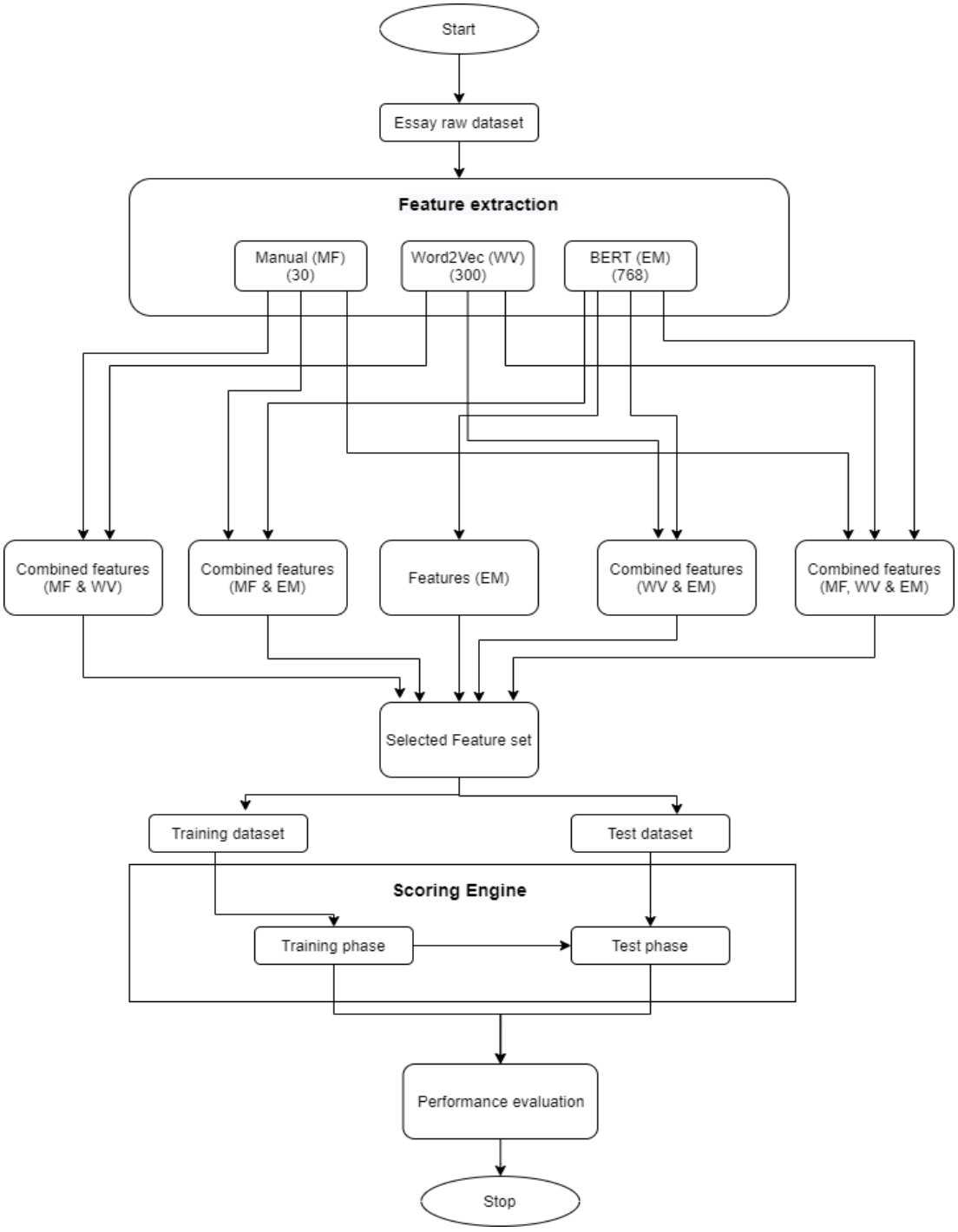

Fig. 1. Experimental Methodology.

TABLE I. SUMMARY OF MANUALLY EXTRACTED FEATURES

Features	Description
Similarity	A similarity measure between 8 manually selected group representatives and the group essays
word count, token count, unique token count	Essay text aggregates
nostop count	Total number of nostop tokens
sentence count	Total number of sentences
ner count	Total number of named entities
Comma count, question count, exclamation count, quotation count	Total number of punctuation entities
organization, caps, person, location, money, time, date, percent	Anonymized entities that were mentioned in the original essay, but were obfuscated before publishing the dataset
noun, adj, pron, verb, cconj, adv, det, propn, num, part, intj	Linguistic entities

TABLE II. HYPER-PARAMETERS OF BERT MODEL

Batch size	16
Optimizer	Adam
Learning rate	1e-4
Dropout	0.7
Model Capacity	4 (128 and 64 hidden units)
Loss	MSE/Categorical Cross entropy
Epochs	200 (reported for the model with best validation loss)
K-fold	10
Cross-Validation Steps	5
BERT model	BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters [7], [8]

TABLE III. FREQUENCIES PER 3-SUBGROUPS IN A CLASSIFICATION PROBLEM

label	0	1	2
Frequency	15.6	34.5	49.9

C. Performance Metrics

This section describes the performance metrics used for measuring the performance of AES models. The most widely used performance evaluation metric is the Kappa statistics, specifically for regression problems. Kappa statistic is an agreement metric whose value ranges from 0 to 1. Kappa statistics can be computed using Equation 1 [8].

$$k = \frac{p_o - p_e}{1 - p_e} \quad (1)$$

Where, p_o represents the observed exact agreement among AES models and p_e represents the hypothetical probability of chance agreement. K=1 indicates that models agree and K=0 indicates total disagreement of AES models. In the case of the classification problem, we measure the performance in terms of accuracy of the AES model. WE computed accuracy from a confusion matrix that gives the number of essays assigned correct score label as expected.

IV. RESULTS AND DISCUSSION

This section presents the experimental results obtained in this work based on given benchmark dataset using different AES models. For a comprehensive comparison of AES models, we use baseline performance as the performance of a combination of 30-manually extracted features and 300-word2vec [23] features reported in the study [24]. In [24], the authors used 330-features and neural network for automated scoring of essays. Furthermore, we use 768 word-embedding features of BERT model. We use combinations of three feature sets to evaluate the performance of AES models. The performance of different models in terms of Kappa statistics and accuracy for the rescaled regression problem and Quantized Classification problem is presented in Table IV.

The values presented for reference model [1] in Table I utilized the 5-fold cross-validation method based on 80% of the dataset in their experiments. The authors of the study [1] have not reported standard deviation estimates. They only reported mean values of Kappa statistic metric. Whereas, in our experiments, we used 90% of benchmark essay dataset. We conducted experiments using 10-fold cross-validation. We presented these results as mean and standard deviation values of Kappa statistics and accuracy for five iterations in our experiments.

In these experiments, we also plotted learning curves for regression and classicization tasks considered in this work based on different feature sets in terms of Mean Squared Error (MSE) and accuracy, respectively. Fig. 2 presents the learning curves obtained in this set of experiments.

It can be observed from Table IV that performance of AES model based on a combination of use 30 manually extracted features, 300 feature dimensional word2vec representation, and 768-word embedding features using BERT model has reported better performance in comparison to the other feature combinations. This model has reported kappa statistics value of 77.2 ± 1.7 for rescaled regression problems and accuracy of 75.2 ± 1.0 for Quantized classification problem. Fig. 3 presents the confusion matrix for the rescaled regression problem based on MF+MV+EM features in this work.

Fig. 4 presents the confusion matrix for the quantized classification problem based on MF + MV + EM features in this work. It can be noted from Table IV that BERT word embedding model has reported the similar performance that of 30-manually extracted features and 300-word2vec features. In the case of BERT embedding, regression problem has better values of Kappa statistics than that of MF-WV combination. In contrast, slightly lower value of accuracy has been reported by BERT embedding for classification problem than that of MF-WV combination. It has been observed that both manual features and Word2Vec embedding methods individually score about 66-67% of accuracy on the quantized classification task. It can also be noticed that WV-EM embedding combination has reported similar performance with minor variation in comparison to MF-WV combination. Such kind of behaviour may be due to the bigger input

dimension size whilst preserving the same model capacity. Small dataset size and curse of dimensionality can be a significant cause of the deceased accuracy of the results. It can also be noted that ME-EM embedding combination of features has reported better-rescaled regression and quantized classification results in comparison to the results reported in [1]. It is worth mentioning that the authors of the study [1] have used an ensemble of LSTM based encoders and XGboost, whereas we employed only a shallow 2-hidden layers feed-forward network.

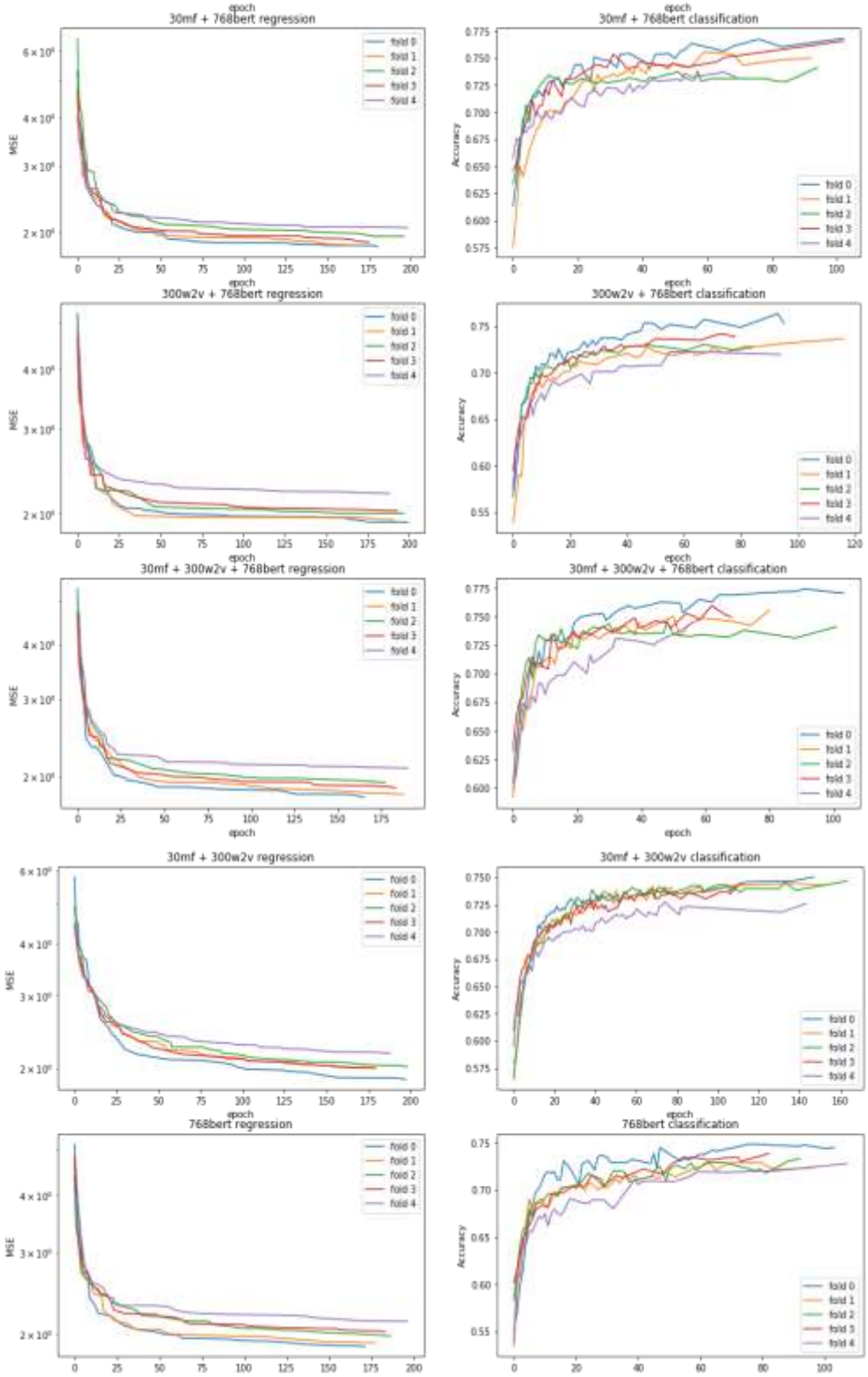

Fig. 2. Learning Curves.

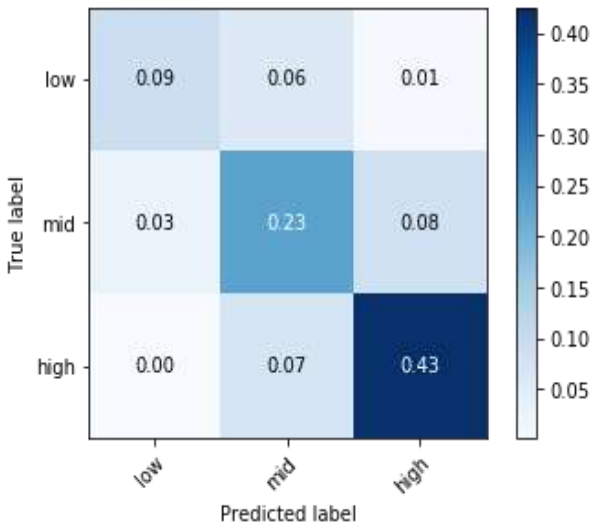

Fig. 3. Confusion Matrix for Rescaled Regression Problem based on MF+MV+EM Features.

TABLE IV. COMPARATIVE PERFORMANCE EVALUATION RESULTS OF AES MODELS

Model	Problem	Parameters (%)	
TSLF-ALL [1]	Rescaled Regression	Kappa Score	77.3
30mf + 300w2v	Rescaled Regression	Kappa Score	74.7 ± 1.5
	Quantized Classification	Accuracy	74.2 ± 0.9
768bert	Rescaled Regression	Kappa Score	76.0 ± 1.6
	Quantized Classification	Accuracy	73.3 ± 0.8
30mf + 768bert	Rescaled Regression	Kappa Score	77.0 ± 1.4
	Quantized Classification	Accuracy	75.1 ± 1.4
300w2v + 768bert	Rescaled Regression	Kappa Score	74.8 ± 1.7
	Quantized Classification	Accuracy	73.5 ± 1.2
30mf + 300w2v + 768bert	Rescaled Regression	Kappa Score	77.2 ± 1.7
	Quantized Classification	Accuracy	75.2 ± 1.0

Nadeem et al. [25] also used BERT embedding for AES. But, they only reported results for the first and second essay groups. Their results are even worse than the results of the AES model based on MF features. They were only able to improve results slightly by using a combination of both feature inputs.

It can be observed from Table IV that the performance of all combinations in case of a rescaled regression problem is better in comparison to the corresponding quantized classification problem. This can happen because a Kappa statistic score is capable of tolerating deviations from a ground truth label and scoring near predictions to some degree. Whereas, accuracy does count only exact category equality.

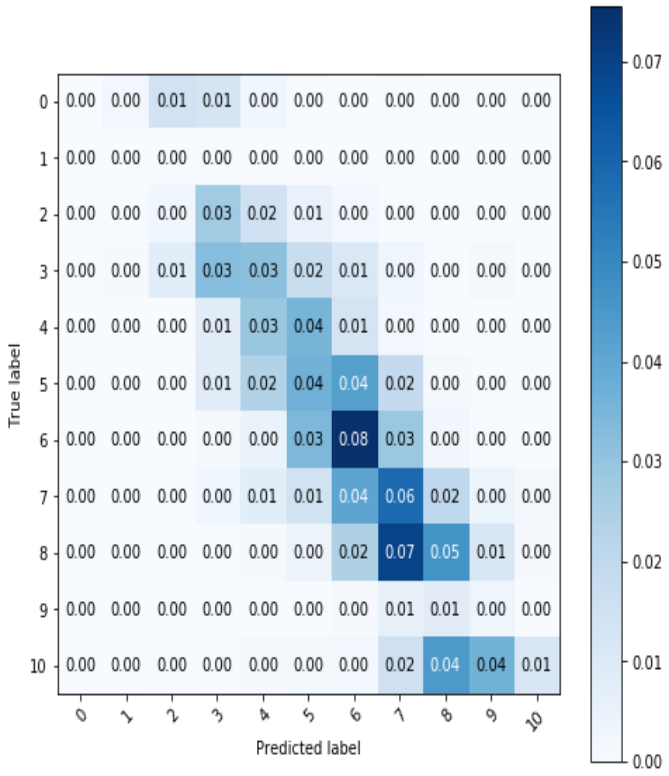

Fig. 4. Confusion Matrix for Quantized Problem based on MF+MV+EM Features Classification.

V. CONCLUSION

Despite many challenges, researchers are investing continuous effects in developing efficient and effective AES using different features of essays. In this paper, we demonstrated a comparative empirical analysis of AES models based on different combinations of various features, namely, manually extracted features, word2vec representation and word embedding using BERT model. The reporting results support our hypothesis that both manually extracted features and deep-encoded features contribute to enhancing the performance of AES models. A combination of manually extracted features, word2vec representation and word embedding using BERT model leads to better performance in comparison to other feature combinations as well as the existing ensemble-based approaches. This combination of features resulted up to 77.2 ± 1.7 of Kappa statistics for rescaled regression problem and 75.2 ± 1.0 of accuracy value for Quantized Classification problem using a benchmark dataset consisting of about 12,000 essays divided into eight groups.

In this paper, we mainly contributed to explaining and comparing AES models based on combinations of various feature sets. We conclude that both manually extracted features and deep-encoded features contribute to enhancing the performance of AES models, makes AES models more reliable than human beings and helps in saving time and money for scoring essays.

REFERENCES

[1] Liu, Jiawei, Yang Xu, and Yaguang Zhu. "Automated essay scoring based on two-stage learning." arXiv preprint arXiv:1901.07744 (2019).

[2] Yannakoudakis, Helen, Ted Briscoe, and Ben Medlock. "A new dataset and method for automatically grading ESOL texts." Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. 2011.

[3] Chen, Hongbo, Jungang Xu, and Ben He. "Automated essay scoring by capturing relative writing quality." The Computer Journal 57.9 (2014): 1318-1330.

[4] Taghipour, Kaveh, and Hwee Tou Ng. "A neural approach to automated essay scoring." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

[5] Alikaniotis, Dimitrios, Helen Yannakoudakis, and Marek Rei. "Automatic text scoring using neural networks." arXiv preprint arXiv:1606.04289 (2016).

[6] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

[7] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[8] Rodriguez, Pedro Uria, Amir Jafari, and Christopher M. Ormerod. "Language models and Automated Essay Scoring." arXiv preprint arXiv:1909.09482 (2019).

[9] Arter, Judith. "Rubrics, Scoring Guides, and Performance Criteria: Classroom Tools for Assessing and Improving Student Learning." (2000).

[10] Page, Ellis B. "The imminence of... grading essays by computer." The Phi Delta Kappan 47.5 (1966): 238-243.

[11] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[12] Ke, Zixuan, and Vincent Ng. "Automated Essay Scoring: A Survey of the State of the Art." IJCAI (2019): 6300-6308.

[13] Chen, Yen-Yu, et al. "An unsupervised automated essay-scoring system." IEEE Intelligent systems 25.5 (2010): 61-67.

[14] Wang, Yucheng, et al. "Automatic essay scoring incorporating rating schema via reinforcement learning." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.

[15] Landauer, Thomas K. "Automated scoring and annotation of essays with the Intelligent Essay Assessor." Automated essay scoring: A cross-disciplinary perspective (2003).

[16] Miltsakaki, Eleni, and Karen Kukich. "Evaluation of text coherence for electronic essay scoring systems." Natural Language Engineering 10 (1) (2004): 25.

[17] Persing, Isaac, and Vincent Ng. "Modeling argument strength in student essays." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015.

[18] Cozma, Mădălina, Andrei M. Butnaru, and Radu Tudor Ionescu. "Automated essay scoring with string kernels and word embeddings." arXiv preprint arXiv:1804.07954 (2018).

[19] Vajjala, Sowmya. "Automated assessment of non-native learner essays: Investigating the role of linguistic features." International Journal of Artificial Intelligence in Education 28.1 (2018): 79-105.

[20] Nguyen, Huy V., and Diane J. Litman. "Argument Mining for Improving the Automated Scoring of Persuasive Essays." AAAI. Vol. 18. 2018.

[21] Rudner, Lawrence M., and Tahung Liang. "Automated essay scoring using Bayes' theorem." The Journal of Technology, Learning and Assessment 1.2 (2002).

[22] Taghipour, Kaveh, and Hwee Tou Ng. "A neural approach to automated essay scoring." Proceedings of the 2016 conference on empirical methods in natural language processing. 2016.

[23] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[24] Turanga1, "Automated students assessment and essay generator", 2012. Online available: https://github.com/Turanga1/Automated-Essay-Scoring. Last accessed on September 13, 2020.

[25] Nadeem, Farah, et al. "Automated Essay Scoring with Discourse-Aware Neural Models." Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2019.