**LITERATURE REVIEW ON AUTOMATIC FEEDBACK GENERATION**

**Prachi Channe1, Shruti Diwate2, Rudragouda Patil3**

12345School of Computer Engineering and technology, MIT Academy of Engineering, Pune

1 [prachichanne820@gmail.com](mailto:prachichanne820@gmail.com) 2 [shrutidiwate555@gmail.com](mailto:shrutidiwate555@gmail.com) 3 [rgpatil@comp.mitaoe.ac.in](mailto:rgpatil@comp.mitaoe.ac.in)

**Abstract**

**Keywords:** Machine learning,Feedback, Grading, NLP(Natural Language Processing)

**1. Introduction**

Essay writing is usually a part of the student assessment process. Several organizations, such as Educational Testing Service (ETS), evaluate the writing skills of students in their examinations. Because of the large number of students participating in these exams, grading all essays is very time consuming. Thus, some organizations have been using Automated Essay Scoring (AES) systems to reduce the time and cost of scoring essays. Automated essay scoring refers to the process of grading student essays without human interference. An AES system takes as input an essay written for a given prompt, and then assigns a numeric score to the essay reflecting its quality, based on its content, grammar, and organization. Such AES systems are usually based on regression methods applied to a set of carefully designed features. The process of feature engineering is the most difficult part of building AES systems. Moreover, it is challenging for humans to consider all the factors that are involved in assigning a score to an essay. Many researches have been done using traditional machine learning algorithms like Linear Regression (LR), Support Vector Regression (SVR), K- nearest Neighbour (KNN) and Decision Tree Classification for grading students’ marks. These models worked well but had certain limitations like huge processing time and lack of correlations in data. Neural Network (NN) will mix input parameters and allow a wide range of possibilities in identifying interdependency in data providing better prediction.

This project aims at developing a model using NN which automatically grades an essay upon submission. Model grades the essay on a scale of 1-10 and the result would be displayed to the user. This system, on the other hand, learns the features and relation between an essay and its score, automatically. Since the system is based on recurrent neural networks, it can effectively encode the information required for essay evaluation and learn the complex patterns in the data through non-linear neural layers. This system is among the first Essay grading systems based on neural networks designed with a combination of different neural networks algorithms layers.

**2. Related Work**

Yali Li and Yonghong Yan [1] developed a system that takes the part-of-speech tag and the words into consideration. Regression was employed in this system instead of classification using the SVM toolbox. It uses the basic model and CVA model for the topic detection component, and can efficiently determine whether an essay is off-topic, especially for huge numbers of essays. In addition, the system is improved by the similarity to full-score essays. Given the two score deviations in comparison to a human rater, it has an accuracy of 86%.

Ben and Hongbo Chen [2] proposed a listwise learning to rank approach to automated essay scoring that incorporates human-machine agreement directly into the loss function. As baselines, this used classical machine learning algorithms, support vector machine (SVM) for classification, regression, and preference ranking, respectively. Experiments on the public English dataset ASAP yielded quadratic weighted Kappas in the 0.80 for prompt-specific rating and 0.78 for generic rating. One of the researchers used the weighted score for the Automated Essay Grading System in the Matrix Laboratory [3]. Using coefficient of determination, it was evaluated how well the K-Nearest Neighbour classifier performed in comparison to the Cosine Similarity Measure in establishing the Weighted Score for the Automated Essay Grading System. For Automated Weighted Score obtained using K-Nearest Neighbor Classifier, R2 = 0.45 and 0.5, while for Weighted Score created with Cosine Similarity Measure, R2 = 0.60 and 0.75.

Three publicly accessible datasets that have been carefully annotated and contain more than 4000 essays on 415 different themes were used in the study's trials [4]. This work used support vector machines, k-nearest neighbour, and a classifier based on linear regression to develop multiple classification algorithms. According to this analysis, Set 1's kNN classifier has a 92.23% accuracy rate with respect to class "B." The most effective classifier for Set 1 is kNN. SVM, which has an accuracy of 78.68%, is the top-performing classifier for Set 7. The study found that the three classifiers had varying degrees of accuracy, but the difference was only about 5%. The accuracy of the results on 4075 essays across 424 different themes and grade range is 73% to 93%.

The NN model for Automated Essay Grading[5] scored 0.94 during the Automatic Essay Grading Kaggle Competition (Neural Networks for Automated Essay Grading, 2016) on the quadratic weighted kappa metric. This neural network model employed a 300-dimensional Glove as the embedding layer initialization. The study on essay scoring [6] demonstrates how a neural network with cross-sentence dependencies and a discourse-based training objective can outperform both feature-based state-of-the-art models and hierarchical LSTMs in terms of automatic essay scoring for the LDC TOEFL essay data. The best results are achieved with a model that learns the combination of hand-crafted features and the neural document representation with Quadratic Weighted kappa of 0.852 and 0.736 for set 1 and set 2 respectively. Research paper on essay scoring presents novel LSTM dependency tree transfer learning scoring method for short essays in Indonesia [7]. The LSTM architecture for essay grading can take both sequence and dependency into account. This proposed technique offers QWK and accuracy results of 53.68% and 16.23%, respectively. The Intelligent Grading System model [8] developed a straightforward grading scheme that achieves a quadratic weighted kappa of 0.7 using machine learning and natural language processing. The tokenized sequences are evaluated using an LSTM neural network, while the vector representation is evaluated using a 2-layer neural network.

The Automatic short answer grading model [9] employed a model to train and test across each collection of essays. Within each essay set, a 5-fold cross validation was performed. The average kappa value for this was 0.73.The SAG [10] uses attention semi-supervised method to train model. It incorporates rule-based model top of regexp matching and span matching methods. Model outperforms with QWK of 0.61 for training size of 200. The ASAG [11] was applied to 9885 answers. Six different approaches were selected to model answers and combined as soft-6. Both linear and quadratic kappa scores were within the range of 0.4 and 0.6. The ASE Transformer Models [12] compares traditional algorithms along with RNN. It implements BERT and regression models with evaluation metrics as accuracy , Cohen’s Kappa of 0.94 and 0.59 respectively for large datasets.

**3. Smart Grading System**

**3.1 Task description**

The objective of this research is to develop a machine learning-based, intelligent system that can grade essays on its own. A dataset with a significant number of essays on a certain topic should be carefully picked in order to ensure consistency among the raters. Our dataset has enough essays on various topics that have been graded. Pre-processing of the dataset is the next step. Cleaning the data is the first stage in the pre-processing process. The process of cleaning data involves removing any inaccurate, incomplete, duplicate, or other wrong data from the dataset. The removal of all characters from the dataset that aren't alphabets is the second step in the data cleaning process. The stop words are then all eliminated from the text. To get rid of stop words, the text is broken up into words, and those words are eliminated if they appear on the NLTK list of stop words.

Word tokenization is then applied to the words. The phrases, sentences, and paragraphs in this passage are divided into many units. Tokens are the name for these more compact objects. Then, these tokens are further examined in order to categorise or count them according to a specific sentiment. Then, using word embedding, we create a Word2Vec model in which the words or phrases are translated into real-number vectors. When words are embedded, those that share a semantically similar meaning are closer together than those that do not. These are then passed to the 3 layered LSTM model. There are a total three models with different layer combinations but common activation function and evaluation metrics. The activation metrics used is ReLU which is explained in the further section of this paper. We have used Quadratic Weighted Kappa as the Evaluation metric for the Models.

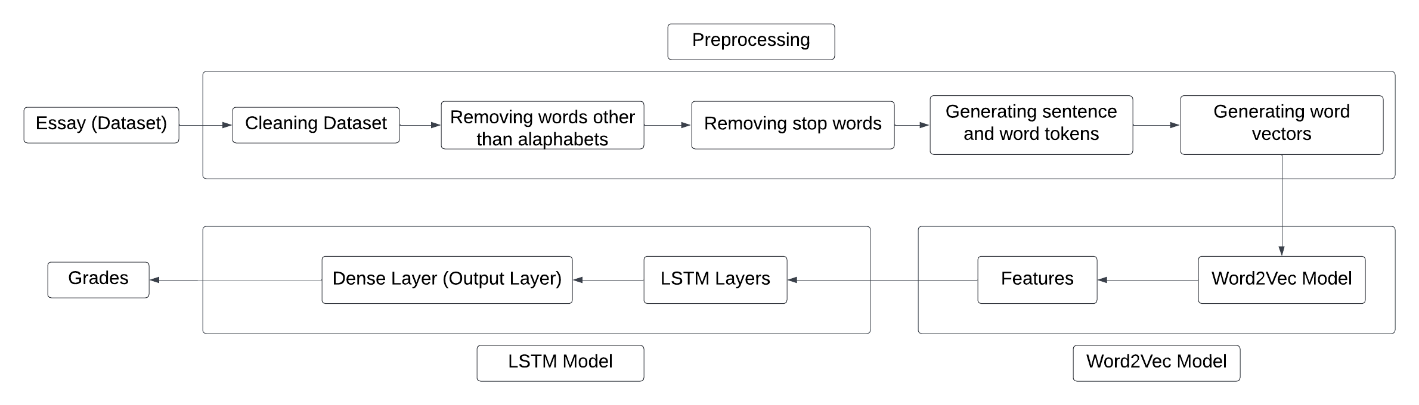


Figure 3.1: Data Flow in the model

**4. System Design**

This section introduces the proposed methodology. We have implemented Recurrent Neural Network (RNN) [20] which are the most commonly used neural network for problem solving by researchers. RNN is a type of Neural Network which uses the output from the previous layer to feed as input to the next layer. Paper consists of implementation and observation of the comparative working of Long Short-Term Memory (LSTM), Bidirectional-LSTM (Bi-LSTM) and Bi-LSTM with attention layer [16]. LSTM is one of the types of RNN which is capable of solving complex problems. This section provides a brief description about the architecture and working of mentioned LSTM models and the training phase.

**4.1 Word2Vec Model (Word embedding model)**

In the Word2Vec model, each given essay consists of a number of words, and each word is represented by a word embedding according to word2vec. The embedding representations are expected to catch the semantic information carried by each word, i.e. the words with similar meanings will be near to each other in the vector space.

For example, from our corpus, we can find that "computer" is similar to "laptop".

The Neural Network can learn to identify the vector of the input word by using the surrounding word of this input. That means if two different words have the same context, the network tends to give them similar word vectors. Hence, every word in our dataset has a unique vector containing the latent semantic and the vectors of the words in one essay can combine to an essay matrix which is the input unit of our scoring machine.

**4.2 3 -Layer LSTM Model**

The paper first introduces the 4-layer LSTM model. It is a sequential model and works better for chosen dataset which is a corpus of essays in text format. The first layer of the model is the word embedding layer. The layer takes 300 as the first argument which is the number of features (output generated from word embedding layer i.e., Word2Vec model), dropout and recurrent dropout as 0.4 respectively and input size from 1 to 300 that is length of each sentence sequence. The next layer takes 64 features as the first argument with recurrent dropout of 0.4. Third layer is also a dropout layer with a dropout value of 0.5. The final layer is a dense layer, it reduces the dimensionality to 1 which is the predicted score. Model have uses ReLU activation function in the dense layer so that the score can be predicted correctly since the values of ReLU function ranges form – ∞ to + ∞. For fitting of training data, models have been passed through batch size of 64 and 100 epochs. These epochs are varied according to the size of the test data. The model produces effective results. Diagrammatic representation is given in figure 4.1. Equations of LSTM are as follows:

Were, is the forget gate, is the input gate, is the output gate, is the cell state, is the hidden state, is sigmoid activation is tanh activation function and ‘.’ is element wise multiplication.

**4.3 LSTM with Bi-LSTM**

To improve the performance / results obtained in Unidirectional LSTM, next we implemented the Bi-LSTM model [26]. Unlike standard LSTM, here, the input is allowed to flow in both directions and it is capable of utilizing information from both sides. Again, it is powerful while modelling the sequential data, improving dependencies between words and sentences in both directions. It adds one more LSTM layer to the previous model, the direction of information flow is revered. Then the model combines the output of both LSTM layers to get the final output. Bi-LSTM is considered to produce more meaningful output when both LSTMs are combined. All other parameters are kept the same. Experiments prove that Bi-LSTM model performs better than the standard LSTM model. Diagrammatic representation is given in figure 4.2. Equations of Bi-LSTM are as follows:

Were, output at time t.

**4.4 LSTM and Bi-LSTM with attention layer**

Though, using Bi-LSTM gave better results, in order to improve model performance, attention layer [16] is introduced to the model. Sometimes, basic LSTM gets confused between the words and can predict the wrong word. So, in order for the encoder to search for the most relevant information, models have been introduced with an additional layer of attention mechanism. By applying the attention mechanism, the model will be able to effectively extract the information between essays through inter-sentence alignment and gain better performance. Diagrammatic representation is given in figure 4.3

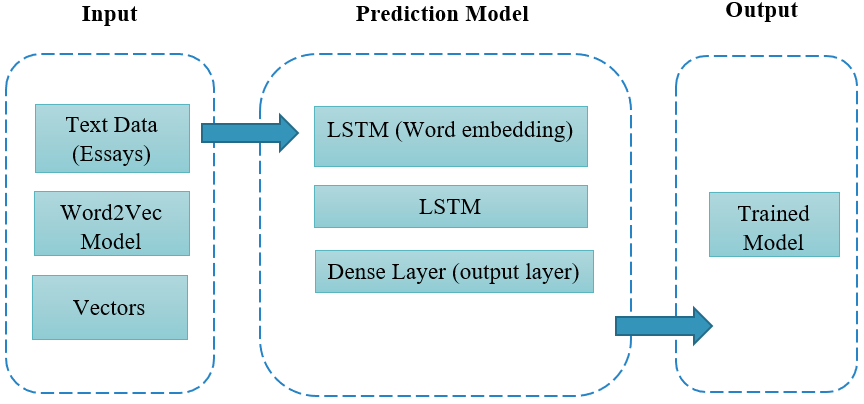


Figure 4.1: LSTM Model

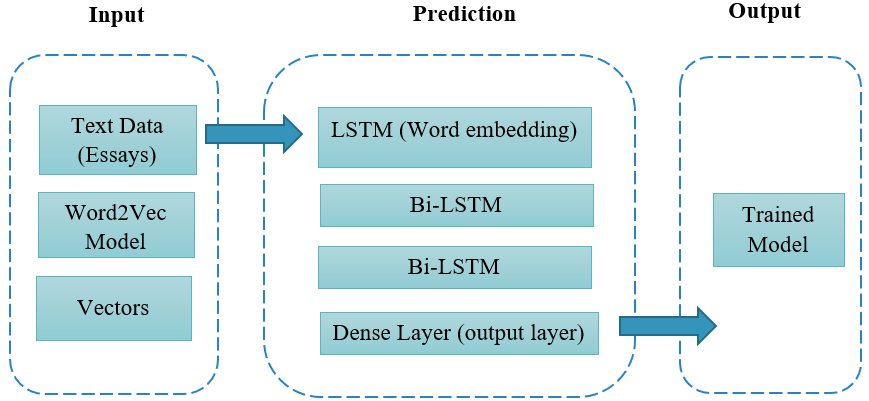


Figure 4.2: LSTM with Bi LSTM Model

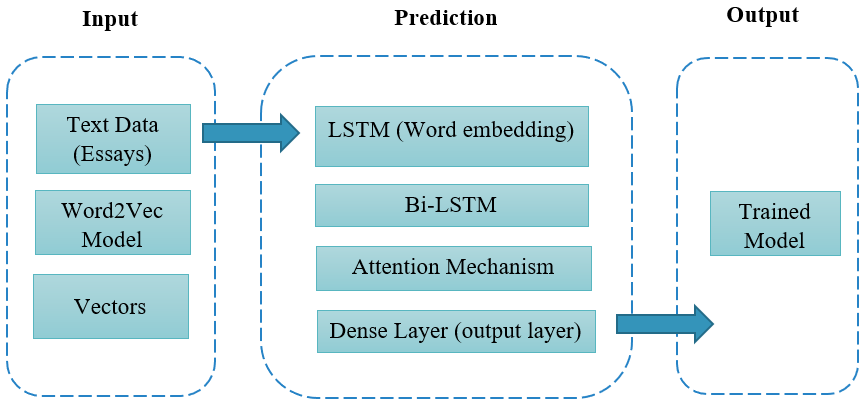
****

Figure 4.3: LSTM and Bi LSTM with Attention

**4.4 Activation Function:**

Layers of nodes make up a neural network, which may be trained to map instances of inputs to outputs. The inputs are multiplied by the weights of a node and added together for a certain node. The node's total activation is known as this value. The activation total is then modified using an activation function, which determines the node's specific output or "activation.". The simplest activation function, where no transform is used at all, is known as linear activation. A network that exclusively uses linear activation functions can be trained relatively quickly, but it is unable to learn complex mapping functions. It is preferable to use nonlinear activation functions since they enable the nodes to understand more intricate data structures. The sigmoid and hyperbolic tangent activation functions are two widely utilised nonlinear activation functions. The sigmoid and tanh functions both saturate, which is a general issue. This indicates that for tanh and sigmoid, high values snap to 1.0 and small values snap to -1 or 0. Additionally, sigmoid and tanh are only really sensitive to input changes around their midpoints, or 0.5 and 0.0 respectively. That’s why we decided to use the ReLU activation function in our model.

ReLU is the most commonly used activation function in machine learning models. Any negative input causes the function to return 0, but any positive value x causes it to return that value. Thus, it may be expressed as:

f(x)=max (0, x)

**5. Experiment Setup and Evaluation Metrics**

**5.1 Setup**

Dataset used in this paper is “The Hewlett Foundation: Automated Essay Scoring” on Kaggle. The dataset includes 8 essay sets. Each of the essay sets was generated from an individual prompt. Average length of essays is in the range of 150 to 550 words per response. All essays were hand graded and were double-scored. The training data is in the format of tab-separated value (TSV) file. There are total 3 scores i.e., rater1 score rater2 score and domain score. The Domain score is an addition of rater 1 and 2 scores. There are some unwanted empty columns also present in the dataset; these columns were dropped using pandas’ libraries. Hence the final dataset has essay\_id, essay\_set, essay, domain1\_score columns.

After generation of the dataset the data needed to be preprocessed before passing on to the model. The essays were first cleaned by removing the stop words, punctuation marks and converting all characters to lowercase. Once the data is cleaned the next step is to generate a feature vector that is to be passed to the word2Vec Model. In order to generate the feature vectors first the cleaned essay was converted into sentence tokens and finally to word tokens. The output of the Word2Vec model was passed as an input to the LSTM layers. For models other than neural network word embeddings were generated.

**5.2 Evaluation Metrics**

Quadratic Weighted Kappa

A set of predictions and a set of multiclass labels are measured by the Quadratic Weighted Kappa index. It attempts to take into account the similarity between the classes, beyond only the class, rather than just focusing on the precision of the match between predictions and labels. As a gauge of agreement between observed raters in cross-classification, Cohen's weighted kappa is frequently used. When ratings are given on nominal scales without an order structure, an appropriate index of agreement is used.Calculation of Cohen’s kappa may be performed according to the following formula:

K = (Pr(a)−Pr(e))/1−Pr(e)

Where Pr(a) represents the actual observed agreement, and Pr(e) represents chance agreement. Generally, a kappa of less than 0.4 is considered poor (a Kappa of 0 means there is no difference between the observers and chance alone). Kappa values of 0.4 to 0.75 are considered moderate to good and a kappa of >0.75 represents excellent agreement. A kappa of 1.0 means that there is perfect agreement between all raters.

MSE

MSE stands for Mean Squared Error. It defines the square of the absolute difference between actual and predicted value. Here, squares avoid the cancellation of negative terms.

MSE = 1/n ( y – y`)2

**6. Results and Discussions**

This section the results of comparison of different machine learning models is explained and further the comparison between different LSTM layers is also proposed. Previous research has been done for comparing models, Table 6.1 shows the results of different machine learning models out of which LSTM gives best performance in terms of QWK as well as MSE. Hence the LSTM model was selected for further experiment purposes and predictions of grades.

|  |  |  |
| --- | --- | --- |
| **Models** | **Kappa Score** | **Mean Squared Error** |
| Linear Regression | 0.85 | 20.94 |
| SVR | 0.64 | 38.19 |
| LSTM | **0.94** | 7.47 |
| K-Nearest Neighbours | 0.92 | 14.7 |
| Decision Tree Classifier | 0.86 | 22.56 |

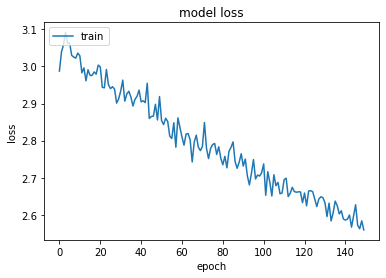
Table no. 6.1: Comparative Study of different machine learning models

Once the model was finalized, various combinations of layers in the LSTM were used for better results. The models were modified with different LSTM, Bi LSTM and Attention layers. The combination of layers used are as follows:

1. LSTM + LSTM
2. LSTM + Bi LSTM + Bi LSTM
3. LSTM + Bi LSTM + Attention

Hyperparameter for all of the models were set as constant with batch size of 64, 50 epochs and activation function as ReLU. 5-fold Cross validation was used in each model. The results of QWK, MSE and variance were noted. The results of this comparative study of layers are given in table 6.2. It is observed that the model that contains the Attention layer performs best amongst all the three models. Further different activation functions were used and the results were noted. The results are shown in table 6.3. For sigmoid and tanh, the QWK is zero and the MSE is very large as compared to the other models. The noted results were not satisfactory and ReLU was decided to be used as activation function for building the final model.

Further different batch sizes were used and the results were noted. The results are shown in table 6.4. The combinations of batch sizes used were 32, 64 and 128. As the standard batch size used is 64 and from our comparison, we found that model with batch size 64 is stable and performs well so we have selected batch size as 64.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Activation Function** | **Epochs** | **Batch size** | **QWK** | **MSE** |
| LSTM (2) | relu | 50 | 64 | 0.94 | 7.86 |
| LSTM + Bi LSTM (2) | relu | 50 | 64 | 0.95 | 6.66 |
| LSTM + Bi LSTM + Attention | relu | 50 | 64 | 0.96 | 6.2 |

Table no. 6.2: Comparative Study of different LSTM Layers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Activation Function** | **Epochs** | **Batch size** | **QWK** | **MSE** |
| LSTM (2) | Sigmoid | 50 | 64 | 0 | 110.45 |
|  | tan h | 50 | 64 | 0 | 109.33 |
|  | relu | 50 | 64 | 0.94 | 7.86 |
|  |  |  |  |  |  |
| LSTM + Bi LSTM (2) | Sigmoid | 50 | 64 | 0 | 124.36 |
|  | tan h | 50 | 64 | 0 | 114.17 |
|  | relu | 50 | 64 | 0.95 | 6.66 |
|  |  |  |  |  |  |
| LSTM + Bi LSTM + Attention | Sigmoid | 50 | 64 | 0 | 114.68 |
|  | tan h | 50 | 64 | 0.95 | 112.23 |
|  | relu | 50 | 64 | 0.96 | 6.2 |

Table 6.3: Using Different Activation function

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Activation Function** | **Epochs** | **Batch size** | **QWK** | **MSE** |
| LSTM (2) | Sigmoid | 50 | 32 | 0.96 | 6.35 |
|  | tan h | 50 | 64 | 0.94 | 7.86 |
|  | relu | 50 | 128 | 0.95 | 7.82 |
|  |  |  |  |  |  |
| LSTM + Bi LSTM (2) | Sigmoid | 50 | 32 | 0.93 | 6.35 |
|  | tan h | 50 | 64 | 0.95 | 6.66 |
|  | relu | 50 | 128 | 0.92 | 6.56 |
|  |  |  |  |  |  |
| LSTM + Bi LSTM + Attention | Sigmoid | 50 | 32 | 0.95 | 6 |
|  | tan h | 50 | 64 | 0.96 | 6.2 |
|  | relu | 50 | 128 | 0.96 | 6.5 |

Table 6.4: Using different batch size

Predictions were made using all the three models generated with selected random essays from the dataset. The results are shown in figure 6.5. It was observed that the model with LSTM, Bi LSTM and Attention layer performs well as compared to the other two models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Essay Id** | **Model** | **Actual Score** | **Predicted Score** |
| 457 | LSTM (2) | 10 | 9 (9.16) |
|  | LSTM + Bi LSTM (2) | 10 | 9(8.558) |
|  | LSTM + Bi LSTM + Attention | 10 | 10(10.113) |

Table 6.5: Predictions

QWK for prediction of all the selected models were calculated. It is shown in table 6.6.

|  |  |
| --- | --- |
| **Model** | **QWK** |
| LSTM (2) | 0.924215844 |
| LSTM + Bi LSTM (2) | 0.956398147 |
| LSTM + Bi LSTM + Attention Mechanism | 0.962121614 |

Table 6.6: Prediction model QWK

**6. Conclusion**

In this paper, we introduced a neural network model with a combination of different layers for essay scoring. For selection of the neural network model various machine learning algorithms were compared and the model with best kappa score was selected. After finalizing the model different combinations of LSTM layers were used to train the model. It was observed that the model with combination of all three layers i.e., LSTM, Bi LSTM and attention gave good QWK and variance. Also, the predictions made by this model were best as compared to the predictions made by the other models. For the selected best model for predictions amongst the three, we calculated the prediction QWK and it was observed that the Model with LSTM, Bi LSTM and attention Layer has the best QWK i.e., 0.96.

**7. Future Scope**

Though our model gives good results with a Kappa Score of 0.96 there is always a scope to improve your model. The Proposed Model works well with the typed essays, further it can be modified for handwritten essays as well as essays in different languages. Not Only essays there are also other modes of question and answers for which this model can be modified for grading. Short answers, long answers specific to questions can also be graded.

**References**

1. Y. Li and Y. Yan, "An Effective Automated Essay Scoring System Using Support Vector Regression," 2012 Fifth International Conference on Intelligent Computation Technology and Automation, 2012, pp. 65-68, doi: 10.1109/ICICTA.2012.23.
2. Chen, Hongbo and Ben He. “Automated Essay Scoring by Maximizing Human-Machine Agreement.” *EMNLP* (2013).
3. Oduntan, Odunayo Esther. “Performance Analysis of K-Nearest Neighbour Classifier and Cosine Similarity Measure in Generating Weighted Scores for an Automated Essay Grading System.” (2018).
4. Goyal, Anjali. (2018). An Empirical Analysis of Machine Learning Models for Automated Essay Grading. 10.7287/peerj.preprints.3518v1.
5. Nguyen, H. (2016). Neural Networks for Automated Essay Grading.
6. Farah Nadeem, H. N. (2019). Automated Essay Scoring with Discourse-Aware Neural Models. Florence, Italy.
7. A. Wiratmo and C. Fatichah. (2020). Assessment of Indonesian Short Essay using Transfer Learning Siamese Dependency Tree-LSTM. 4th International Conference on Informatics and Computational Sciences (ICICoS).
8. Zhu, Wilson & Sun, Yu. (2020). Automated Essay Scoring System using Multi-Model Machine Learning. 109-117. 10.5121/csit.2020.101211.
9. Manvi Mahana, M. J. (2012). Automated Essay Grading Using Machine.
10. Wang, T., Funayama, H., Ouchi, H., & Inui, K. Incorporating Rubrics to Short Answer Grading.
11. Galhardi, L., de Souza, R. C. T., & Brancher, J. (2020, November). Automatic grading of Portuguese short answers using a machine learning approach. In *Anais Estendidos do XVI Simpósio Brasileiro de Sistemas de Informação* (pp. 109-124). SBC.
12. Ludwig, S., Mayer, C., Hansen, C., Eilers, K., & Brandt, S. (2021). Automated essay scoring using transformer models. *Psych*, *3*(4), 897-915.
13. Jan Philip Bernius, Stephan Krusche, and Bernd Bruegge. 2021. A Machine Learning Approach for Suggesting Feedback in Textual Exercises in Large Courses. Proceedings of the Eighth ACM Conference on Learning @ Scale. Association for Computing Machinery, New York, NY, USA, 173–182. DOI:https://doi.org/10.1145/3430895.3460135
14. Urs Von Matt, “Kassandra:The Automatic Grading System”,Institute for Scientific Computing,ETH Zurich,Switzerland,1994
15. Z. Wang, J. Liu and R. Dong, "Intelligent Auto-grading System," 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2018, pp. 430-435, doi: 10.1109/CCIS.2018.8691244.
16. Neslihan Süzen, Alexander N. Gorban, Jeremy Levesley, Evgeny M. Mirkes, Automatic short answer grading and feedback using text mining methods, Procedia Computer Science, Volume 169, 2020, Pages 726-743, ISSN 1877-0509,
17. Aldea, A. I., Haller, S. M., & Luttikhuis, M. G. (2020). Towards grading automation of open questions using machine learning. In J. van der Veen, N. van Hattum-Janssen, H-M. Järvinen, T. de Laet, & I. ten Dam (Eds.), *Engaging, Engineering, Education: Book of Abstracts, SEFI 48th Annual Conference University of Twente (online), 20-24 September, 2020* (pp. 584-593). University of Twente.
18. R. Krithika and Jayasree Narayanan. 2015. Learning to Grade Short Answers using Machine Learning Techniques. In Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15). Association for Computing Machinery, New York, NY, USA, 262–271. DOI
19. Dr. A. Mercy Rani 2021. Automated Explanatory Answer Evaluation Using Machine Learning Approach. *Design Engineering*. (Jun. 2021), 1181 - 1190.
20. Sanchez, Victor & Nerbonne, John & Verspoor, Marjolijn. (2013). Identifying Important Factors in Essay Grading Using Machine Learning..
21. Liu, T., Ding, W., Wang, Z., Tang, J., Huang, G.Y., Liu, Z. (2019). Automatic Short Answer Grading via Multiway Attention Networks. In: Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., Luckin, R. (eds) Artificial Intelligence in Education. AIED 2019. Lecture Notes in Computer Science(), vol 11626. Springer, Cham. <https://doi.org/10.1007/978-3-030-23207-8_32>
22. Ormerod, C. M., Malhotra, A., & Jafari, A. (2021). Automated essay scoring using efficient transformer-based language models. *arXiv preprint arXiv:2102.13136*.
23. Sharma, Aman & Urkude, Vaibhav & Sonawane, Adhoksh & Sharma, Nidhi. (2021). SMART ESSAY GRADER SYSTEM. 3. 2582-5208.
24. Chauhan, R. K., Saharan, R., Singh, S., & Sharma, P. (2020). Automated content grading using machine learning. *arXiv preprint arXiv:2004.04300*.
25. Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289*.
26. Ludwig, S., Mayer, C., Hansen, C., Eilers, K., & Brandt, S. (2021). Automated essay scoring using transformer models. *Psych*, *3*(4), 897-915.
27. Chauhan, R. K., Saharan, R., Singh, S., & Sharma, P. (2020). Automated content grading using machine learning. *arXiv preprint arXiv:2004.04300*.