Descriptive Answer Script Grading System using CNN-BiLSTM Network

Shirien K A, Neethu George, Surekha Mariam Varghese

Abstract—Descriptive answer script assessment and rating program is an automated framework to evaluate the answer scripts correctly. There are several classification schemes in which a piece of text is evaluated on the basis of spelling, semantics and meaning. But, lots of these aren't successful. Some of the models available to rate the response scripts include Simple Long Short Term Memory (LSTM), Deep LSTM. In addition to that Convolution Neural Network and Bi-directional LSTM is considered here to refine the result. The model uses convolutional neural networks and bidirectional LSTM networks to learn local information of words and capture long-term dependency information of contexts on the Tensorflow and Keras deep learning framework. The embedding semantic representation of texts can be used for computing semantic similarities between pieces of texts and to grade them based on the similarity score. The experiment used methods for data optimization, such as data normalization and dropout, and tested the model on an Automated Student Evaluation Short Response Scoring, a commonly used public dataset. By comparing with the existing systems, the proposed model has achieved the state-of-the-art performance and achieves better results in the accuracy of the test dataset.

Keywords: Artificial Intelligence, Convolution Neural Networks, LSTM Networks, Machine Learning

I. INTRODUCTION

Examinations play an important role in the education system by determining the intellectual ability of the students. After analysis, the teachers expend much of their time for assessing the marks of the students and the assessment requires bulk use of human energy, time and expense. In some cases due to the bulk of available students in specific subjects ,it requires more teachers effort for answer script scoring ,Thus an automated evaluation program will reduce the effort in assessment process. Today several automated assessment systems [1] [5] exists and they evaluate a piece of text based on grammar, spelling and meaning. The assessment of concise answers remains an open question. Their effectiveness is a major problem among existing systems. Hence, more reliable and effective automated descriptive response script evaluation method has been proposed for evaluating the descriptive response scripts. The evaluation of answer paper is more dreary process that requires great effort from evaluators. The unavailability of expert evaluators and more number of answer sheets to be evaluated makes the evaluation process more time consuming. Existing automated text evaluation systems [1] [2] analyze a piece of text based on semantics, spelling and context. But many of them are not efficient. The main aim of this project is to extract the semantics to efficiently represent the texts in answer scripts and develop a model from the key and evaluated answer scripts to grade non evaluated answer scripts using deep learning. The instinctive nature of the answer scripts evaluation leads to variations in grades awarded by different human assessors, which is anticipated by students as a great source of negligence. This difficulty of grading answer sheets may be faced through the automated answer script evaluation tools. An automated assessment system must be capable of scoring the answer papers within the range of those awarded by human evaluators. It must be persistent in the way it scores and thus it can save the time and cost of evaluation. Currently there exists many automated essay evaluation systems based on keyword matching, sequence matching and using bag of words model. Nowadays, many researchers focus on the agile and most meticulous area of machine learning, i.e the deep learning. By introducing an automated assessment system using deep learning can improve the efficiency of answer evaluation. The objective of this project is to excerpt the semantics to efficiently represent the text in answer scripts and develop a model from the key and evaluated answer scripts to grade non evaluated answer scripts using deep neural networks. Deep Neural networks are able to capture semantics of text in order to find the similarity between texts. The goal of the system is to replace the traditional human evaluation of the answer sheet that depends on several factor such as time, mindset, presentation style and so on. A peaceful approach for evaluating the response script using deep learning is suggested here. It is a hybrid of machine learning and NLP. The learning is done using cells from the Convolution Neural Network and LSTM. The embedding vector which corresponds to the last word is the semantic representation of the whole sentence. For anticipating the marks of answers a specified model is trained which consists of an embedding layer, Convolution Layer, Bi-LSTM layer, dropout layer and fully connected neural network layer.

Manuscript received on January 07, 2021.
Revised Manuscript received on January 15, 2021.
Manuscript published on January 30, 2021.
* Correspondence Author
Shirien K A*, Computer Science Department, Mar Athanasius College of Engineering, Kothamangalam, India. Email: shirienashraf2211@gmail.com
Neethu George, Computer Science Department, Mar Athanasius College of Engineering, Kothamangalam, India. Email: neethugeorge777@gmail.com
Dr. Surekha Mariam Varghese, Computer Science Department, Mar Athanasius College of Engineering, Kothamangalam, India. Email: surekh.var@gmail.com

II. LITERATURE SURVEY

In 1966 Ellis Page developed the first AES system, Project Essay Grader, at the request of the American College Board. The PEG system defines a large set of surface text features from essays, it fails to detect an essay's content related features and ignores the semantic aspect of essays and focuses more on surface structures.

E-rater, developed in the late 1990s by Educational Testing Services (ETS) in America, is a commercial AES device that was put into practical use in the GRE (Graduate Record Examination). The E-rater device uses the natural language processing techniques to collect various types of essay linguistic features such as lexical, syntactic, grammar, etc. Then it calculates the final score by the stepwise regression process. The classification-based approach sees essay scores as indiscriminative class labels and uses classical classification algorithms, e.g. the K-nearest neighbor (KNN) and the naive Bayesian model, to determine to the group an essay belongs, wherever a group is linked to a numeric ranking. Intelligent Essay Assessor (IEA), jointly developed in the late 1990s, evaluates essay by measuring semantum characteristics. - ungraded essay, represented by a semiconductor created by Latent Semantic Analysis (LSA), is classified according to the degree of similarity with graded essay semiconductors. Latent semantic analyzes are used to calculate the similarity of cosine between unscored essays and scored essay. LSA is a statistical model of word use that empowers similitude of semantic similarity between textual knowledge items. The key point is the meaning of a passage is very much dependent on its terms and it may result in meaning changes in the passage by changing only one word.[1].

In 2001, Callear et al. pursued a further line of research leading them to the Automated Text Marker (ATM). ATM searches for concepts and their dependencies in the text to finally give two independent scores, one for the style and one for the content.[2]. In 2003, Rose et al. introduced CarmelTC, a hybrid text classification method for evaluating essay responses to qualitative physics questions, offers another point of view. It classifies text pieces based on the features extracted from a syntactic analysis, as well as the classification of the same text by Naive bayes[3]. Philip E. Robinson et al.[4] subsequently presented an online learning platform for programming teaching, learning and evaluation in 2017.The online learning platform open source helps in the teaching and evaluation of computer programming in large classes. The paper explains the technology and application of the learning framework and new approaches for automated assessment of programming assignments and exams. Jamsheedh et al.[1] have introduced a simple response script evaluation system by using NLP and machine learning software. Here, features are extracted from the humanly evaluated sample dataset of response scripts and the answer key given for high weight. Model Bag of Words are used for representing the features extracted. The system was capable of grading unevaluated paper. The system not concerned with checking synonyms and antonyms, and more erroneous. It evaluates scores for just the exact sentence found in the word bag. The output of the system in many cases larger than the expected output and evaluates marks more than once for the same type of answer. Panchami et al.[5] used Support Vector Machines to implement a diagram evaluation method. The program used machine learning, image processing, and natural language

processing to interpret diagrams found in response texts in an efficient way. The proposed method tests diagram effectively by considering the text present in the diagram, the syntax of the text, the form of blocks and the direction of arrows. The program aims to be a stepping stone in the diagram evaluation area. Charusheela Nehete et al.[2] followed another line of research that led them to the Checkpoint, an application for descriptive response checking and grading based on the NLP. The online assessment system stores the answers as text files after the processing assessment. This has a substantial execution time and cannot accommodate very complex sentence structures. First, it analyzes the response given to it as an input. Considering the existence of synonyms it will test how close the answer given is to the ideal answer whose keywords the teachers will provide. It will grade the answers according to similarity. It uses answer scripts generated by the online assessment. Pantulkar S et al.[6]have explained a paper that computes the similarity between sentences based on word net. They show that measuring semantic similarity of sentences is closely related to semantic similarity between words. Here, three different semantic similarity approaches like cosine similarity, path based approach (wu palmer and shortest path based), and feature based approach are evaluated and tested. The preprocessing of pair of sentences identifies the bag of words and then the similarity measures are applied based on wordnet.Ming Che Lee et al.[7] have presented a paper that contains grammar and semantic corpus based similarity algorithm for natural language sentences. The sentence similarity algorithm takes advantage of corpus based ontology and grammatical rules to overcome the addressed problems. Peipei Li et al.[8]have proposed an efficient and effective approach for semantic similarity using a large scale semantic network. This semantic network is automatically acquired from billions of web documents. Atish and Vijay[9] have presented a paper based on calculation of semantic similarity between two words, sentences or paragraphs. The algorithm initially disambiguates both the sentences to ensure the right meaning of the word for comparison. To calculate the semantic similarity between words and sentences, the proposed method follows an edge-based approach using a lexical database. The information content from a corpus can be used to influence the similarity in particular domain. Semantic vectors containing similarities between words are formed for sentences and further used for sentence similarity calculation. Word order vectors are also formed to calculate the impact of the syntactic structure of the sentences. Since word order affects less on the overall similarity than that of semantic similarity, word order similarity is weighted to a smaller extent. In the published work of Yuhuva et al.[10]the emphasis is specifically on measuring the similarity between very short sentence-length texts. They were presented with an algorithm that takes the semantic information and word order information implied in the sentences into account. Two sentences semantic similarity is measured using data from a structured lexical database and corpus statistics. Using a lexical database enables our method of modeling knowledge of common sense in humans and integrating corpus statistics enables our approach to be adaptable to various domains.

Deep structured semantic model DSSM[11]and Convolutional structured models CLSM[12], developed for information retrieval, sentence embedding methods. However, DSSM treats the input sentence as a bag-of-words and does not model word dependencies explicitly. CLSM treats a sentence as a bag of n-grams, where n is defined by a window, and can capture local word dependencies.

Then a Max-pooling layer is used to form a global feature vector. These models, by design, cannot capture long distance dependencies, i.e., dependencies among words belonging to non-overlapping n-grams. Long short-term memory networks were developed in[13] to address the difficulty of capturing long term memory in RNN. It has been successfully applied to speech recognition,[14][15] which achieves state-of-art performance. Nitish et al.[16]have been implemented a regularization technique to resolve overfitting. The main concept of dropout is to randomly drop units (along with their connections) from the neural network during preparation. That prevent too much co adaptation by units. Dropout samples from an exponential number of various "thinned" networks during the preparation. At the time of testing, it is easy to approximate the effect of averaging all these thinned networks' predictions by simply using a single unthinned network that has smaller weights. This greatly eliminates overfitting and provides major advantages over other methods of regularization.

III. PROPOSED METHOD

The proposed model estimates scores for detailed answers using CNN and Bi-LSTM. The system takes the whole short answer as input and converts it into a glove vector represented by the use of the embedding layer. Bi-LSTMRNN learns temporary data; the embedding vector is from the embedding layer and corresponds to the final vector semantic representation of the whole answer. It is given as input to the dropout layer and then to the neural network layer that is fully connected with the soft-max activation function. The last layer predicts the score. The problem in the beginning is developing an effective model estimate the scores of the answer scripts. Changed to suit problem a recent trend in research, in-depth practice. Problem when moving from solution domain, three different versions of the system were designed and developed using CNN and the Bi-LSTM neural network. All of these have different modules, The models are identical and are described in the following sections. There are three model systems: CNN component, Bi-LSTM component and CN-BiLSTM component in comprehensive learning.

The system consist of three modules namely- Preprocessing, Sentence Embedding and Grading. The proposed model performs preprocessing as the first step. The dataset is the input to the preprocessing module. The dataset format is [question ID], [key ID],[answer Id], [answer], [mark].The performance measures for the proposed system are evaluated relevant to the performance parameters training and testing percentage.

The proposed model uses GloVe vectors for representing the answers. Glove allows for parallel implementation, it's

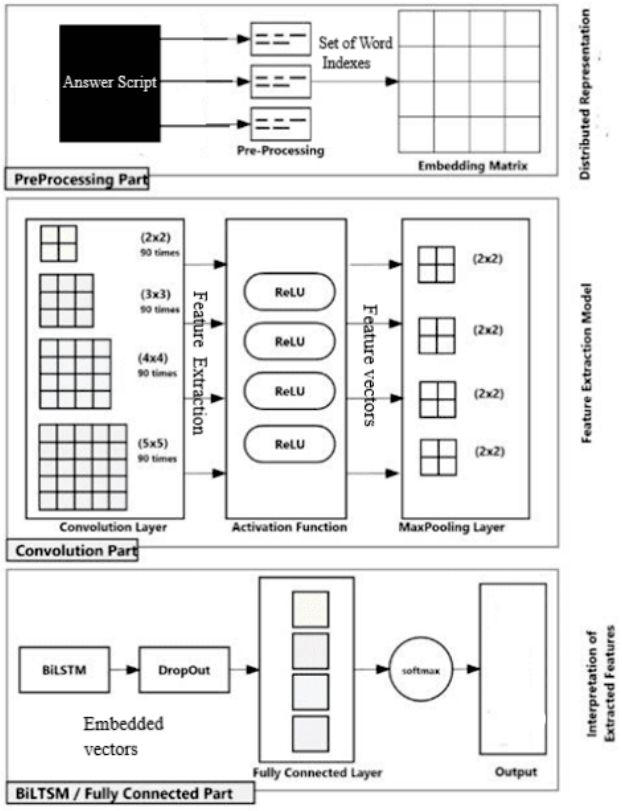

Fig. 1. System Architecture

easier to train over more data. It focuses on semantic similarity by converting each word to Glove vector representation in the vector space. The model sequentially takes each Glove vector of the word and converts it into the semantic representation, the embedding vector corresponding to the last word will be the semantic representation of the entire descriptive answer. During training, features are extracted to create a model from human evaluated sample dataset of answer scripts based on a key. The evaluated sample dataset contains answers and their corresponding score. This model is used to efficiently represent semantics of the answers as embedding vectors. The embedding vector corresponding to the last word will be the semantic representation of the entire short answer. The model consists of an embedding layer, Convolution Layer, Bi-LSTM layer, dropout layer and fully connected neural network layer.

A. Preprocessing

This stage extracts the relevant features of the text in the answer scripts and converts them into glove vector indexes. A dataset is prepared from the key and answer scripts with its corresponding human evaluated score. Glove.6B contains pre-trained word vectors of Wikipedia 2014 and Gigaword 5 that are used for obtaining the vector representation of the words in the sentences. Glove is an unsupervised learning algorithm for developing word vector representations. A word-id dictionary is created from the glove vector dataset

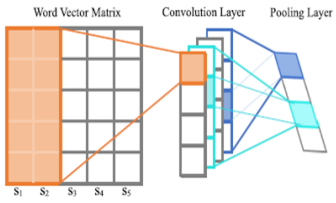

Fig. 2. CNN Part in CN-BiLSTM Model

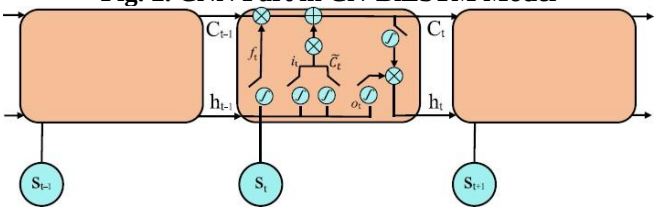

Fig. 3. LSTM Part in CN-BiLSTM Model

with id as its index to the corresponding glove vector of the word. The features of the answers are extracted after tokenization and lemmatization. The tokenization converts a sentence into a set of words. By doing lemmatization, the lemma of each word is obtained. For each set of lemmatized words of an answer word-id dictionary lookup is performed which results a set of glove vector ids and are used for semantic extraction. Also we need to convert the corresponding score for each answer in the dataset into a form that can be used by the proposed sequential model. The scores can be converted to any of the vector representation such as one hot vector.

B. Semantics Representation Using CN-BiLSTM

The next step is to extract the meaning of the answers given by the students. A one-dimensional convolutional neural network is used in the CN-BiLSTM layer to capture a sentence's structural data. Through the convolution operation, the convolution kernel tests the text matrix for sliding scanning, performs function extraction and feature selection, and finally combines the sentence expression vectors. Therefore, at different locations, a set of characteristics are acquired, including information between words and structural information formed between adjacent words. The structural features of the phrases used in the model are increasing, as the above operations are repeated many times with different convolution kernels. Multiple vector representations are obtained by the final model, and then these vectors are related to obtain the entire sentence's high-level semantics. Formula(1) is a one-dimensional convolutionary formula, as shown (examples with 3 convolution kernels).

$$H=F(X(t-1),X_t,X_(t+1) =W[X_(t-1),X_t,X_(t+1)]+b \quad (1)$$

The coded information that passes through the CN layer is sent to the Bi-LSTM network. The Bi-LSTM is essentially the chronological order of the integration filtering information. LSTM is a special network structure with "forget gate," "input gate," and "output gate. In order to allow information to selectively affect the state of each moment in the recurrent neural network, LSTM relies on the structure of certain "gates." As follows, the basic formulas are:

$$C_t = f_t * C_(t-1) + i_t * C'_t \quad (2)$$

$$f_t = \sigma(W_f \cdot [h_(t-1), x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i \cdot [h_(t-1), x_t] + b_i) \quad (4)$$

$$C'_t = \tanh(W_c \cdot [h_(t-1), x_t] + b_c) \quad (5)$$

which f_t, i_t represents forgetting door and input door respectively. The forgotten gate controls the degree of oblivion of the previous moment at each moment; the input gate controls the degree to which new memories are written into long-term memory. The value of sigmoid function is in [0, 1]; tanh function is in [-1, 1]. $C_(t-1)$ is the state at time t-1, and C_t is the state at time t. –

$$h_t = o_t * \tanh(C_t) \quad (6)$$

$$o_t = \sigma(W_o \cdot [h_(t-1), x_t] + b_o) \quad (7)$$

where O_t is the output gate, which controls how short-term memory is affected by long-term memory. H_t is the output of t. The Bi-LSTM is based upon the LSTM structure. There are two intermediate values being stored. In the forward calculation, A participates and A' participates in the reverse calculation. The final performance value of h is based on A and A'. In the forward calculation, the C_t of the hidden layer is related to $C_(t-1)$. In the reverse calculation, the C_t of the hidden layer is related to $C_(t+1)$:

$$C_t = F(U_{xt} + WC_t - 1) \quad (8)$$

$$C'_t = F(U'_{xt} + W'C_t + 1) \quad (9)$$

$$h_t = g(V C_t + V' C'_t + 1) \quad (10)$$

Where g is the output activation function U, V, W, and U ', V', W 'is the matrix of the parameters for forward and backward calculation. A series of word structure vectors completed by each convolution kernel of a one-dimensional convolutionary network is the Bi-LSTM semantic representation modeling. Each position has an intermediate representation, the semantics of the end of the phrase to this location, which represents the beginning of the clause to this position. Here, we assume that the intermediary representation of each position is determined by the vector of the current position's word structure, the intermediate representation of the previous position, and the intermediate representation of the next position. The outcome is input to the output layer after going through a Bi-LSTM.

IV. RESULT AND PERFORMANCE EVALUATION

The Data collection for the assessment was compiled manually from 112 separate response scripts. The dataset is split into a training set and a test set and includes a total of 10 prompts for questions. The word embedding aspect is 64 in experiments, the drop out ratio is 0.3, the number of convolution kernels in a one-dimensional convolution network is 64, the convolution kernel size is 3, the convolution phase size is 1, and the bidirectional LSTM size is 64, the last layer of fully linked layer units is 38, the activation function is softmax, the optimizer is adam prop, and the loss function is categorical cross-entropy, and the keras framework is used to measure it. The predictions must have predicted probabilities for each of the class models.

The cross-entropy is then summed across each features and averaged across all examples in the dataset. The Quadratic Weighted Kappa is used as the evaluation metric for finding the agreement between the grades predicted by model and the human graders. It calculates the level of agreement between the two raters. It also takes into account the chance probability of assigning the same grade to a sample by both the raters. Quadratic Weighted Kappa is calculated as follows. Firstly, the weight matrix M is constructed according to Equation 11. Here i is the reference rating of human rater, j is the rating assigned by the model and N is the total number of possible ratings. The selection of the accurate score is done by considering the standard deviation of scores. Testing shows that the accuracy directly depends on the richness and diversity of answers in the train data. The experiments have been conducted with different random samples of data and average accuracy of each of the models in each 100 iteration has been calculated.

$$M_{i,j} = (i-j)^2/(N-1)^2 \quad (11)$$

$$K = 1-(\sum i,jM_i,jO_i,j)/(\sum i,jM_i,jE_i,j) \quad (12)$$

Here matrix O contains the observed scores such that rating i is given by human grader and j is given by the model M_i, j contains the weights as derived in Equation 11 and E contains the expected scores obtained by multiplying the histogram vectors of the two scores i.e. the ones by human graders and the other by the proposed model. Sub-scripts in Matrix O_i, j correspond to the number of essays that score i from the first rater and j from the second one. The experimental setup was initially done with 112 re- sponse scripts. Further performance evaluations are done on the basis of short answer scoring dataset by the Hewlett Foundation. It can be known that the final outcome of the CNN model is low and the accuracy on the test set is not stable by means of the accuracy rate change graphs and the final experimental results on the training set and the test set. Furthermore, the CNN model converges quicker and the training of the model takes less time since a large number of parameters are shared by the convolution operation. The final accuracy is not good as the volume and operation only provide the sentence's internal structural information, but it is not possible to extract the sequence information convolution operation. LSTM network can get a better result compared to the former two networks, but the convergence speed is sluggish. On the training set, the BiLSTM network provided good results, but the test set had bad results, suggesting that the model had significant overfitting. The CN-BiLSTM network balances the model training tools to boost accuracy while meeting the way the human body gets textual knowledge in its own way. That is, the text information structure is the object of attention when reading; when the text semantics are carefully understood, the sequence information is more important.

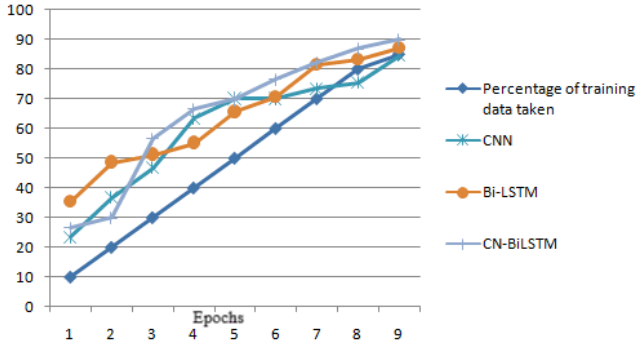

Fig. 4. Training percentage V/S Accuracy

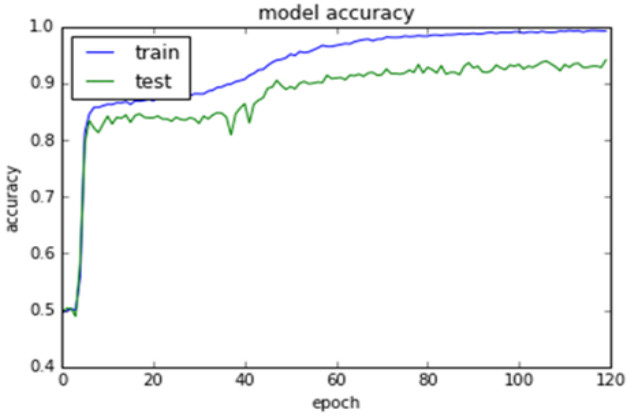

Fig.5. Training and Testing V/S Accuracy

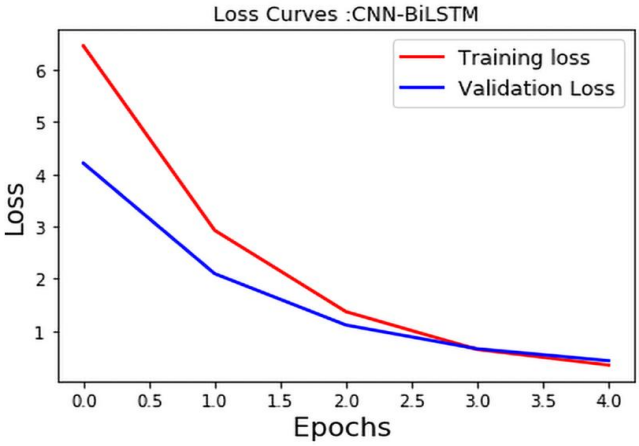

Fig.6. Training and Validation V/S Accuracy

Table 1.Training Data V/S Accuracy Percentage Of Various Models

Accuracy Percentage of Various Models			
Training Percentage	CNN	BI-LSTM	CN-BIL STM
10	23.38	35.52	26.66
20	36.66	48.68	30
30	46.66	51.31	56.6
40	63.33	55.26	66.66
50	70	65.78	69.9
60	69.99	70.81	76.66
70	73.33	81.57	82.36
80	75.33	83.31	86.99
85	84.42	87.26	90.04

V. CONCLUSION

. In this paper, CNN and BiLSTM worked together to build a model to greatly enhance the accuracy of the response script grading process.CNN performs better with lower training percentages. But, CNN-BiLSTM model outperforms both CNN and BiLSTM in accuracy when training percentage of 50 or more is chosen. CNN-BiLSTM model enhances flexibility for encoding text information and captures the context's long term dependencies well. The experimental results indicate that this approach increases the accuracy and shows the viability in implementations of the deep learning model. CNN pays more attention in extracting deep local characteristics. Application of CNN in response scoring reveals that CNN is close to human vision, focusing on local knowledge and combining local features at last. RNN is still the basis for the simulation of grading systems demonstrated in various studies because of its advantages in sequence modeling. The RNN modeling concept adheres to the method of producing languages, and several challenges are solved by combining various techniques. The current method is an automated program that uses deep learning to screen and rate descriptive responses. The Natural Language Processing and Deep Learning techniques have been used to describe and rate concise responses semantically. The data collection for the assessment was compiled manually from 112 separate response scripts. Features are extracted from the human evaluated sample dataset to construct a answer script model and high score given key. This model outperforms current approaches and achieves an accuracy of about 90.04 % accuracy.

REFERENCES

1 Jamsheed C.V, Aby Ababai T and Surekha Mariam Varghese, "A fair assessment system for evaluation and grading of text in Digitised Descriptive Answer Scripts, "RACCCS, vol. 4, pp. 346-352, 2016.

2 S. U. Charusheela Nehete, Vasant Powar and J. Wadhwani, Wadhwani, Checkpoint An Online Descriptive Answers Grading Tool, IJARCS, vol. 8, pp. 135- 155, 2017.

3 A. Deshpande, Deep learning research review week 358 natural language processing., Available at https://adeshpande3.github.io /adeshpande3.github.io/Deep Learning-Research-Review-Week-3 -Natural -Language-Processing.

4 M. Kiser, Introduction to Natural Language Processing (NLP)-Algorithmia Blog., Available at https://blog.algorithmia.com/ introduction-natural- language-processing-nlp/.

5 Panchami K.S, Surekha Mariam Varghese and Aby Abhahai T, Grading of Diagrams in Answer Scripts Using Support Vector Machine, International Journal of Control Theory and Applications, vol. 10, pp. 345-350, 2017.

6 P. B. Tom Mitchell, Terry Russell and N. Aldridge, Towards robust computerised marking of free-text responses, In Proceedings of the 6th CAA Conference, Loughborough, 2002.

7 C. Godbout, Recurrent neural networks for beginners-camron godbout–medium, Available at https://medium.com/@camron godbout/recurrent-neural-networks–for–beginners -7aca4e933b82.

8 K. Q. Z. Z. W. X. H. Peipei Li, Haixun Wang and X. Wu, A Large Probabilistic Semantic Network based Approach to Compute Term Similarity, vol. 2, pp. 1041-4347, 2015.

9 A. Pawar and V. Mago, Calculating the similarity between words and sentences using a lexical database and corpus statistics, IEEE Transactions on Knowledge and data Engineering, February 2018.

10 Z. A. B. J. D. O. Yuhua Li, David McLean and K. Crockett, Sentence similarity based on semantic nets and corpus statistics, vol. 2, pp. 1041-4347, 2015.

11 J. G. L. D. A. A. P.S. Huang, X. He and L. Heck, Learning deep structured semantic models for web search using clickthrough data, in Proceedings of the 22nd ACM International Conference on Information Knowledge Management, pp. 2333-2338, 2013.

12 J. G. L. D. Y. Shen, X. He and G. Mesnil, A latent semantic model with convolutional pooling structure for information retrieval, CIKM '14 Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 101-110, November 2014.

13 S. Hochreitur and J. Schmidhuber, Long short-term memory, Neural Computation, vol. 9, pp. 1735-1780, November 1997.

14 F. B. Hasim Sak, Andrew Senior, Long short-term memory recur- rent neural network architectures for large scale acoustic modeling, in Proceedings of the Annual Conference of International Speech Communication Association, 2014.

15 A. M. Alex Graves and G. Hinton, Speech recognition with deep recurrent neural networks, in Proc. ICASSP, Vancouver, Canada, May 2013.

16 A. K. I. S. Nitish Srivastava, Geofrey Hinton and R. Salakhutdinov, Dropout:A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research, pp. 929-1958, 2014.

AUTHORS PROFILE

Shirien K A received a Bachelor of Technology in Computer Science and Engineering from KMEA Engineering College, Edathala in 2018 and received Master of Technology in Computer Science and Engineering from Mar Athanasius College of Engineering , Kothamangalam affiliated with APJ Abdul Kalam Technological University in 2020. Her research interest is in Deep Learning and Data Mining.

Neethu George received Master of Technology in Computer Science and Engineering in 2019 from Mar Athanasius College of Engineering , Kothamangalam affiliated with APJ Abdul Kalam Technological University and received Bachelor of Technology in Computer Science and Engineering from Government Engineering College Idukki in 2017 . Her research interest is in Deep Learning and Data Mining.

Dr. Surekha Mariam Varghese currently heads the Computer Science and Engineering department, Mar Athanasius College of Engineering, Kothamangalam, Kerala, India. n 1990, she received her B-Tech Degree in Computer Science and Engineering from the College of Engineering, Trivandrum affiliated with the University of Kerala, and M-Tech in Computer and Information Science from the University of Cochin, Kochi, in 1996. In 2009, she earned a Ph.D. in Computer Security from the University of Science and Technology in Cochin, Kochi. She has some 27 years of experience in teaching and research at various institutions in India. Her research interests include machine learning, network security, database management, algorithms and data structures, operating systems, and distributed computing.