An Optimized Approach for Automated Short Answer Grading using Hybrid Deep Learning Model with PSO

[1]S. Ganga, [2]Prof. S. Sameen Fatima

[1]Research Scholar, [2]Professor
Dept of Computer Science and Engineering,
University College of Engineering(A), Osmania University, Hyderabad, India

***Abstract*— In recent years, research into automatic grading has accelerated significantly. Right now, there has never been a greater need for an effective ASAG system. The start of the pandemic and the switch to online learning have given the research even more momentum. Several methods have been suggested by authors from around the world to resolve the ASAG task. The purpose of this work is to demonstrate how models based on Transfer Learning and optimization methods using swarm intelligence can be utilized to grade short answers. In the proposed research, we introduce a short answer grading system using BERT, BiLSTM, CNN with PSO optimization. To increase the performance of the scoring systems, we optimize the input features using PSO and then given to the hybrid deep learning based model consisting of the BERT, BiLSTM and CNN to classify the answers as correct, partially correct and incorrect. The model is tested using the base line data set i.e., Mohler data set as well as a newly created Computer Science data set in Indian context (CSDSIC). Various experiments are conducted to test the performance of the model. The performance metrics used for evaluating the model are accuracy and root mean squared error. Model shows an accuracy of 92%.**

***Index Terms*—Bidirectional Encoder Representation from Transformers (BERT), Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (Bi-LSTM), Particle Swarm Optimization (PSO), Automated Short Answer Grading (ASAG)**

I. INTRODUCTION

In the educational system, Fundamental elements for assessing the effectiveness of the learning process include assessing and measuring the knowledge gained. Because of the lengthy process of the evaluation task, it is required to make the assessment more efficient while maintaining or even raising its quality [1], for example by utilizing some level of grading automation. Since many years, computerized evaluation has been used at school and college level, but mostly for multiple-choice questions that require recognition of choice made. Such recognition-based question as per research are flawed because they don't adequately account for many facets of learned knowledge, like self-explanation and reasoning [2]. As a result, open-ended recall questions that ask students to create their own solutions are more frequently employed in academic settings. Gap filling answers and essay-style answers are not considered in the present research; rather, focus is in short answers which are made up of few words or few sentences [3].

Due to variations in linguistics, the same answer may be expressed in many ways, the nature of grading i.e., there may be multiple correct answers or wrong answers, and the lack of consistency in human rating i.e., on an ordinal scale within a range, grading student written short answers using instructor-provided reference answers is a challenging task in natural language understanding. Aside from the use of word embeddings for ASAG task [4] and use of neural networks for essay grading [5], the neural models based on transfer learning and usage of swarm optimization techniques have not been applied extensively for ASAG despite state-of-the-art results in the various natural language processing tasks.

In this research, we introduce a unique ASAG system that consists of a Optimized feature selection using PSO, followed by hybrid neural layers of BERT, Bi-LSTM, CNN for predicting the classification label for a given student answer based on the reference answer.

Particle Swarm Optimization (PSO), is a heuristic-based stochastic searching method created by Kennedy and Eberhart [6]. PSO algorithm is a form of searching process which is on the basis of swarm, where every item is known as a particle defined in D-dimensional search space as a potential solution of the optimized problem, and it is capable of recalling that of its own and the optimal position of the swarm, in addition to the velocity.

BERT [7] is a multi-layer bidirectional Transformer encoder method for Natural Language Processing (NLP). Google team has created the pre-trained BERT with unlabeled data which was taken from books with 800 million words and Wikipedia with 2,500 million words. Fine-tuning an extra classification layer and all of the pre-trained parameters, the pretrained BERT model can be used for specific tasks of NLP [8].

The context information is captured with the use of BiLSTM, which is a combination of LSTM units and bidirectional recurrent neural network (BiRNN) models. This is done in order to ensure that every moment contains the context information. On the input sequence, an LSTM is applied during the first round of processing (i.e., forward layer). During the subsequent iteration, the LSTM model is presented with the mirror image of the input sequence (i.e., backward layer). Applying the LSTM twice results in better long-term dependency learning, which enhances the model's accuracy [10].

Convolutional Neural Networks (CNN's) were first proposed for image classification tasks, and they performed remarkably well in tasks which require object detection. It is comparable to a traditional neural network that includes a series of operations. Generally, these processes include: convolution and pooling. [11] has suggested a CNN variation for tasks involving natural language processing that uses 1D convolution and pooling operations.

II. Related Work

The ASAG task has been in research for many years. There is a need to improve the grading quality and also generalizing the task to different domains. There have been many works in this area, but need a more reliable and efficient method for automatic grading. The birth of automatic grading systems started with automating the essay grading systems [19]. The essay grading systems have achieved state-of the art achievements but these are stilled to be achieved for short answer grading. Burrows [3] has initially classified the various existing ASAG systems in 5 categories based on the timeline of the proposed development and the techniques used. ASAG systems are broadly divided into 2 types, i.e., features based machine learning based applications and then with advanced developments in technology, deep learning-based applications were developed. Presently many researchers are working on application of transformer-based models for ASAG [20].

The word embeddings based on deep learning methods are used to extract the semantic representations of the text. In [21], word vector representations derived from Word2Vec [22] and GloVe [23] were used to generate the embeddings and then similarity measures were applied for those representation. Many methods asl combined the hand-crafted features along with the deep learning-based word embeddings to improve the performance.

In [25], a deep learning framework for ASAG is introduced based on BERT. Here, student answer and reference answer are encoded using a BERT model which is a pre-trained model. This model is fine-tuned for solving the text classification problems with small corpus. Then a semantic refinement layer is built to enhance the semantics of the BERT outputs, using a bidirectional-Long Short-Term Memory (LSTM) network and a Capsule network with position information in simultaneously. This allows to develop a potent semantic representation of answers. In a next step, a triple-hot loss technique is applied for prediction task in ASAG.

In [24], an iterative method is built as an ensemble model which consists of a classifier of student written responses and uses various similarity measures-based model classifier. Also feature extraction of the unlabeled data is handled using the transfer learning-based approaches of co-relation analysis

In [26], a variety of methods for producing vector representations of student responses such as Sentence-BERT as well as traditional approaches such as Word2Vec and Bag-of-words are applied. Unlike the other approaches, this method has considered the question, its context, rubric information is also modelled. Model is tested for out-of-sample generalization.

In [27], it proposed that by fine-tuning a pre-trained model like BERT with self-attention can be applied to short answer grading and can in turn be used for grading datasets of different domains. It is possible to achieve higher accuracy in grading using this model.

III. Proposed Methodology

The ASAG problem can be described as Regression problem where the model is trained to predict a grade of the student written answer based on similarity between the reference answer and student answer. The ASAG problem can also be described as a Classification problem, where the model is trained in such a way to classify the given student answer as correct answer, partially correct answer or incorrect answer based on the semantic representation of the student answer and the reference answer. There could be even more classes defined as required. The present research implements the Classification of the student answers as correct, Incorrect or partially correct answers.

The steps involved in this process are:

1. Data Augmentation
2. Optimized Feature set generation with PSO
3. Building a Hybrid Deep Learning Model
4. Training the Model
5. Testing the Model

A. Data Augmentation Strategy:

The present dataset contains a smaller number of (reference answer, student answer) training pairs in our dataset, deep learning techniques might not be completely supported. We suggest a data augmentation technique for short answer scoring that utilizes the correct student responses in order to lessen this issue. In this study, we make the assumption that the right student response is an additional type of reference response to supplement our training dataset. Here, we outline our approach to data augmentation. Data augmentation uses existing data to create modified copies of datasets, which are then used to artificially increase the training set. It involves making little adjustments to the dataset or creating new data points using deep learning.

According to the teacher's review of our statistics, a sizable portion of student responses receive correct scores. We hypothesize that the student answers that obtained correct scores by the teacher are equivalent to teacher-provided reference answers since there are many students whose answer label is correct. Following this, one of the correct student answers which is not very similar to the reference answer is chosen as second reference answer. By doing so, the dataset is doubled than the original dataset.

B. Optimized Feature Set with PSO:

An initial population of particles makes up the PSO. It is assumed that every particle is a point in an N-dimensional space. Every particle position will be indicated as Li that is a set of $l_{i1},l_{i2},l_{i3}\ldots,l_{iN}$ and the corresponding Velocity is represented as V_i that is a set of $v_{i1},v_{i2},v_{i3}\ldots,v_{iN}$. Pbest indicates the best preceding location of any particle and is provided by $PL_i=(pl_{i1},pl_{i2},pl_{i3}\ldots,pl_{iN})$. Gbest is the index of the global best particle. Every particle chooses its trajectory during the optimization process based on both its Pbest and Gbest so far. In the canonical PSO, the update rules of position and velocity are defined as

$$V_{id} = iw*V_{id}+a1*rand() *(PL_{id}-L_{id})+a2*rand()*(pad-L_{id})$$

$$L_{id} = L_{id} + V_{id}$$

In which iw is the inertia weight, a1 and a2 will be the acceleration coefficients referred as "self-cognitive" and "social learning". They determine the learning weights for Pi and Gbest. Rand() is used to generate a random number in the range[0,1].

The input features are extracted from the Bert tokenizer. These input features are grouped into different sets using the 10-fold cross-validation. Then the PSO searches for the best configuration while generating a set of random solutions. The accuracy of the classification and the quantity of chosen features are used to calculate each of the configuration's quality. The personal best and global best solution are used to update solutions in the next iterations and are repeated until the stopping conditions are met.

Fitness Function: Based on the outcomes of the cross-validated set, the canonical distribution of the log loss/cross entropy function is setting up by the fitness function at each iteration of the PSO's genome sequencing. The following is the formula for cross entropy and logarithmic loss:

$$\text{Entropy(t)} = \sum_{j=1}^{M}[y_j log(p(y_j))] \qquad (1)$$

$$Log\ Loss = -\frac{1}{N}\sum_{1}^{N}[y_i log(p_i) + (1-y_i) log(1-p_i)] \qquad (2)$$

The chance that an item will be chosen in the loop is shown above by the symbol $p(y_j)$. N is the total number of things, while y_i denotes the item's output. 1 has a probability of pi, while 0 has a probability of p_i. In place of general fitness score, the new hybridised algorithmic framework utilizes the log loss validation curve and seeks for improving the distribution of the feature space through fitness optimization for each individual feature. Feature selection's major problem is to reduce the fitness loss brought on by the PSO approach.

Global and Local Data Execution Trajectory per Epoch is given in Fig 3.1. This Graph Represents the trajectory per epoch for how much time(y) is taken to find an optimum point(X) locally [in 1 cluster] and globally in entire data. The objective value decreases to zero as the iteration increases for both global objectives chart and local objectives chart. Global and Local Best Fitness graph is given in Fig 3.2. The graphs show the probability of finding Best Optimum Points per iteration. The fitness function value reduces to zero for both global and local best fitness as the iteration increases. Number of selected optimal features: 36

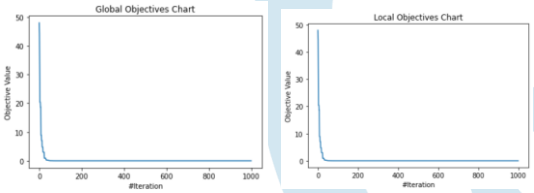

Fig: 3.1 Global and Local Data Trajectory per Epoch

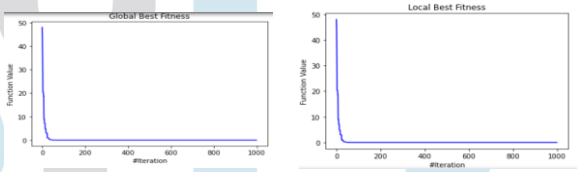

Fig 3.2 Global and Local Best Fitness

C. Hybrid Deep Learning Model.

Use Pre-trained BERT contextualized representations have achieved cutting-edge results on a range of downstream NLP tasks by fine-tuning task-specific data. However, we argue that certain conventional neural networks, including Capsule networks as well as Bi-LSTM, will retrieve particular adequate semantics for a given input sequence that cannot be acquired by the transformer, realizing the mutual promotion and semantic complementarity between the classical NN and the BERT design. Initially, a word's hidden state is encoded by BERT transformers using the weighted sum of all other word embeddings in order to preserve resources [12]. This method fully exploits the connections between all words but ignores the order and distance. Due to the absence of temporal data in BERT encoding, it is required to first build finer global context information using memory cells and different gate architectures in Bi-LSTM. For every hidden state in BERT, convolutional kernels are used in CNNs to extract the pertinent local context. Next, BERT may provide dynamic word embeddings for its top network that can change from sentence to phrase. Large-scale unsupervised learning is used to pretrain BERT, and downstream tasks are used to refine it. In comparison to more traditional static word embeddings like Glove and ELMO, the BERT's dynamic word embeddings have more all-purpose information [13], which helps with the convergence and training of neural networks' upper layer. The conventional neural network on BERT can therefore operate well with less corpus [14], [15]. Finally, studies have shown that combining the upgraded BERT model with traditional neural networks can increase performance in specific specialized tasks with a limited number of training corpora like sentiment analysis. For instance, Yang et al. [18] have suggested a multihead consideration on a fine-tuned BERT prototype to sum up the distance weights for Chinese-based aspect polarity classification, Liao et al. [16] coupled RoBERTa [17] with CNN to enhance the precision of aspect-category sentiment estimation, Nguyen et al. [19] stacked CNN on upper edge of BERT for the forecasting of extraction data in domain-specific business with limited information. Last but not least, optimum selection is necessary to lessen computational complexity and produce a consistent interpretation from textual attributes. By lowering the covariance in textual data, the gradient and fitness functions aid in the convergence to their optimal values and the loss caused by feature covariance.

As per the aforementioned discussion, we propose a novel method where in the input features are optimized using the PSO optimizer and then given to the BERT model. The output of the Bert layer is a then input to the Bi-LSTM and CNN to capture the global and local context respectively. The architecture of the model is given below (Fig 3.3).

As explained in the PSO section above, the input to the PSO is the list of all the features. The text input features of student answer and the reference answer taken using BERT tokenizer are given as input to the PSO algorithm. The feature selection method seeks to identify the optimum subset of features that are essential for class prediction. If the irrelevant set of features is used, a model might not be able to accurately represent the answer. Because irrelevant attributes have a negative impact on the performance of

the model, the choice of pertinent characteristics is an essential component of model development. Instead of using a dynamic embedding-only strategy, BERT layer uses a fine-tuned method.

Although the parameters of BERT layer will all be initialized based on the pretrained BERT model, BERTBASE [18], they need to be collaboratively adjusted with the model's other layers. The BERT token encodings, of which CLS is the first token and SEP is the delimiter of sentence pairs, correspond to the tokens in the students' as well as reference answers. The preliminary outputs of the BERT layer-processed input sequence (i.e, the last hidden states of the BERT) are now processed using the Bi-LSTM layer and the CNN layer. These layers are used for further refinements of the semantics. We use CNN network's convolutional layer to excerpt relevant local context for the BERT layer's hidden states. We retrieve BERT outputs' fine global context using complex gate designs in the Bi-LSTM network. Both CNN and Bi-LSTM network are active at the same time. Layer normalization is performed on the output of these layers and the combined representations are now input to the multihead attention layer. In this layer, every head will be a scaled dot product of Key (K), Query (Q), and Value (V) vectors.

The output representation of the multihead attention is now given to the prediction layer. Maxpooling of the output gives the last semantic input representation. Then a linear transformation and softmax is computed to predict the output class. Dropout is added at the prediction layer for preventing overfitting. The cross-entropy loss is used for updating model weights while training. For categorizing jobs in the case of ASAG, one-hot multinomial distribution is used. It converts the answer pair representation Z into a distribution of one-hot gold vector.

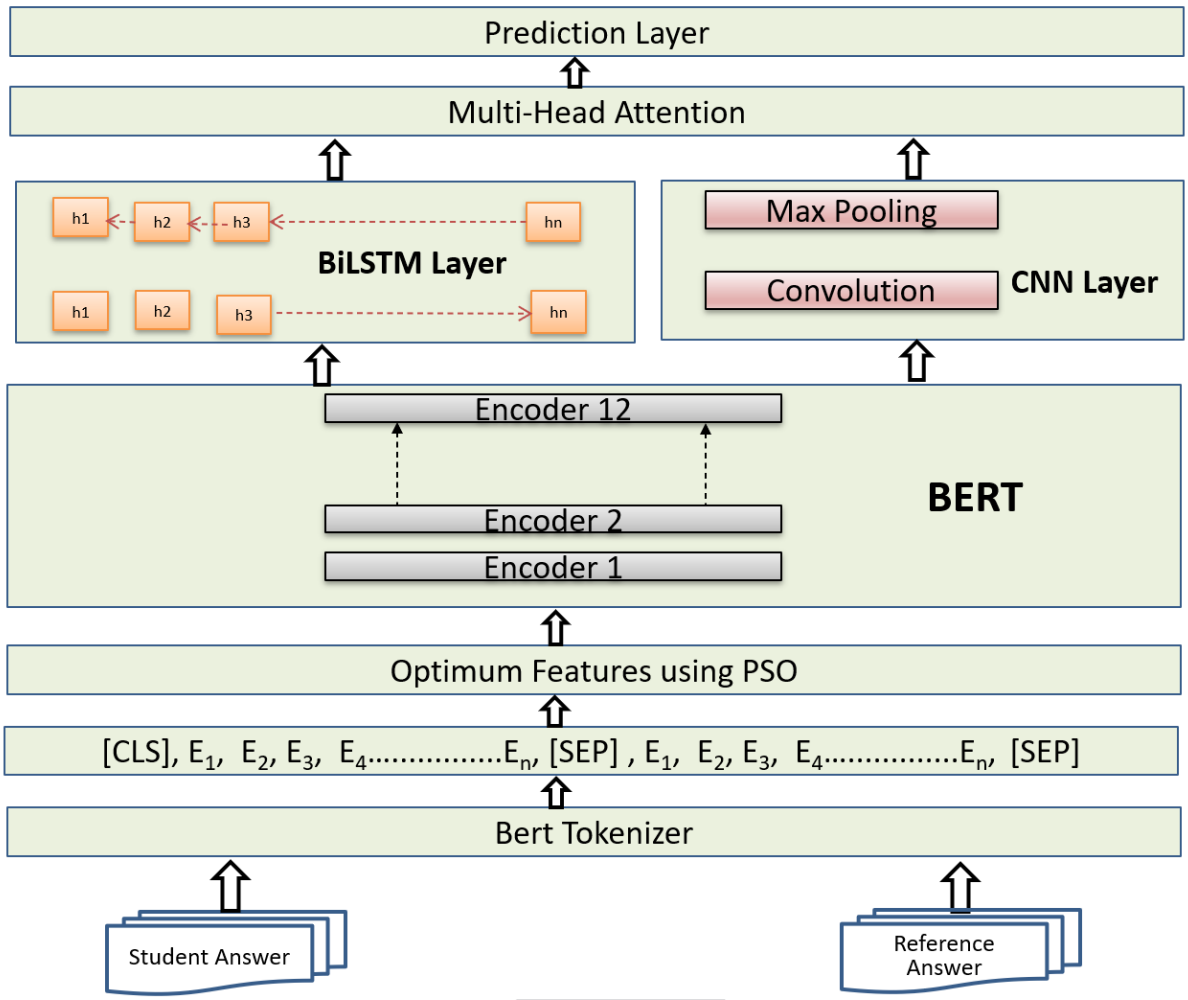

Fig-3.3 Proposed Model Architecture

IV. EXPERIMENTS

1. *Dataset:*

The proposed model is tested using two datasets:

A. Mohler ASAG Dataset

Mohler ASAG is a baseline dataset and is widely used for developing models for ASAG tasks. In an introductory computer science course taught at the University of North Texas, Mohler et al. constructed a dataset of computer science short-answer questions using the ten assignments and two tests of the course. It has a total of 80 questions with the responses of 2273 students. Two teachers independently rated the responses of each student using a numeric scale ranging from 0 to 5. The true score of the students' answer was determined by taking the average of the two scores that were labelled, which resulted in 11 scoring grades ranging from 0 to 5 with 0.5 intervals between each grade. The sample of the dataset is given in figure 38. The shape of the dataset is (2273, 7). It has 2273 rows and 7 columns. The columns have question, desired answer, student answer, scores.

B. Computer Science SAG Dataset in Indian Context (CSDSIC)

Computer Science SAG Dataset in Indian Context is a newly created dataset for the purpose of ASAG. This dataset is created by collecting the graded answer scripts of the class tests in various Engineering colleges in our city. The answer scripts of computer science courses such as operating systems, Database systems, software engineering, data structures and machine learning were taken. The answers are digitized along with the question, score assigned. The corresponding reference answers are provided by the teachers. Each of the answers is labelled as correct, partially correct or incorrect based on the marks scored. This dataset is peer reviewed by two teachers who teach these subjects. The total number of questions are 44 with each question having around 30 to 50 answers. The total number of answers is 1740. Since the dataset size is small, we have done data augmentation as described above. This increased the data set from 1740 to 3480 rows. Sample dataset is given in Table 4.1. The number of answers after the split into train-test ratio for both the datasets and the train test split of the augmented CSDSIC is given in Table 4.2.

S.No	QUESTION	Student Answer	max mar	Marks Obtain	Class Lab	Reference Answer
4.3	What is Belady's anomaly ?	The number of page faults changes as the frame size increases this is known as Belady's anomaly.	1	0.5	1	Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.
4.3	What is Belady's anomaly ?	In some situations while performing page replacement algorithm increase of page numbers increase the number of page faults (i.e decrease hit ratio) is called Belady's anomaly. It occurs in first in first out page replacement algorithm.	1	1	2	Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.
4.3	What is Belady's anomaly ?	Belady's anomaly means the number of hits in the FIFO (first in first out)than the faults then it is called Belady's anamaly.	1	0	0	Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.
4.3	What is Belady's anomaly ?	Belady's anomaly means it states that the file search into the sequance should go out in the sequance ways. It is called Belady's anomaly	1	0	0	Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.
4.3	What is Belady's anomaly ?	Belady's anomaly: whenever through we increase the of frames in the page replacement techniques, the performance decreases sometimes we may occur more no of page faults.	1	1	2	Belady's Anomaly is the phenomenon of increasing the number of page faults on increasing the number of frames in main memory.

Table- 4.1 Sample answers from Computer Science SAG Dataset in Indian Context.

	Datasets	
	CSDSIC	Mohler
Train	1392	1818
Test	348	455
Total	**1740**	**2273**

	Augmented CSDSIC
Train	2784
Test	696
Total	**3480**

Table 4.2 Train Test split of both data sets and Train Test split of Augmented CSDSIC.

2. *Implementation*

The experiments are run on Google Colab with GPU. Niapy python package is used for PSO, Tensorflow, Keras and Python are used for implementation.

Both the data sets are divided in the ratio of 80:20 for training and testing. As the first step, the input to the system, i.e., student answer and reference answer are tokenized using the Bert tokenizer. The output is a set of features that describe the student and reference answers. These features are input to the PSO.

The PSO will fetch the least optimum and best optimum points local and globally, for which the initialization will be 1000 swarm population (10 group pop size * 100 iteration size), and PSO's flags, min, max will remain the same as its default values, while the problem statement will contain lower and upper bound values, fitness function used is the log loss function as described above. During the optimization it will keep track of global and local best optimum point based on each iteration which will also help us to find out the best features afterwards, After the optimization the optimized data will be send to BERT's Input layer.

The pretrained $BERT_{BASE}$(uncased_L-12_H-768_A-1) with 12 layers, 768 heads, 110M parameters are used. The maximum length of the sequence is 128, with word embedding size is 300 and batch size of 64 The number of units in LSTM are 200, activation function is relu, dropout is 0.1. The size of each convolution cores is set to 3 and the number of convolution cores in CNN is set to 400, dropout is set to 0.1. The model is trained with PSO optimized data, a learning rate of 2e-5 and a batch size of 64 for 10 epochs.

V. RESULTS

Fig 5(a) shows the how the loss is decreased with a greater number of epochs for both test and training data of the CSDSIC dataset and Fig 5(b) shows the increase in the accuracy as the number of epochs are increased.

Table 1 gives the comparative analysis for Computer Science SAG dataset in Indian context. The proposed hybrid model using BERT, BiLSTM, & CNN with PSO optimizer is giving the highest accuracy of 95%. Other comparisons are done like BERT, BiLSTM, & CNN with ADAM optimizer and BERT, BiLSTM, & CNN with SGD optimizer are considered. Analysis using BERT, BiLSTM with Adam optimizer, BERT, BiLSTM with PSO optimizer is also considered.

Table 2 gives the comparative analysis for Mohler dataset. The proposed hybrid model using BERT, BiLSTM, & CNN with PSO optimizer is giving the highest accuracy of 92%. Other comparisons are done like BERT, BiLSTM, & CNN with ADAM optimizer and BERT, BiLSTM, & CNN with SGD optimizer is considered. Analysis using BERT, BiLSTM with Adam optimizer, BERT, BiLSTM with PSO optimizer is also considered.

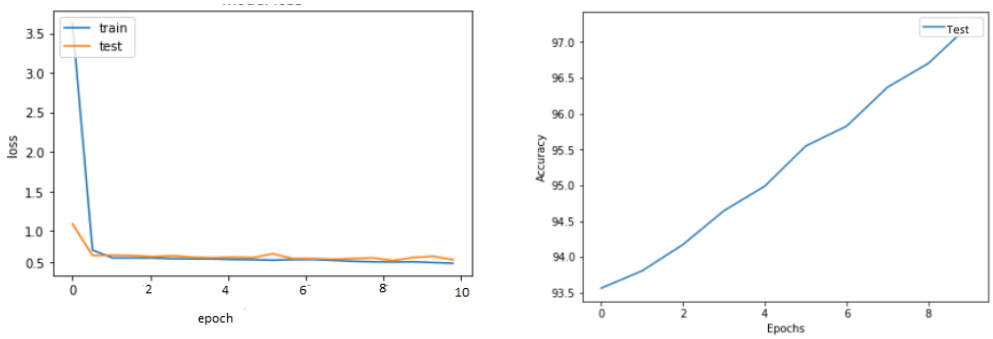

Fig 5(a) Loss Vs epoch and (b) Accuracy Vs epochs for CSDSIC Dataset

Model	Accuracy	Mean Absolute Error
BERT+BiLSTM+CNN with Adam Optimizer without PSO	0.78	0.613
BERT+BiLSTM with Adam Optimizer without PSO	0.71	1.13
BERT+BiLSTM+CNN with SGD Optimizer without PSO	0.73	0.945
BERT+BiLSTM with SGD Optimizer without PSO	0.69	1.32
BERT+BiLSTM with PSO Optimizer	**0.84**	**0.918**
BERT+BiLSTM+CNN with PSO Optimizer	**0.92**	**0.497**

Table 1 Comparative Evaluation on Computer Science SAG Dataset in Indian Context

Model	Accuracy	Mean Absolute Error
BERT+BiLSTM+CNN with Adam Optimizer without PSO	0.76	0.627
BERT+BiLSTM with Adam Optimizer without PSO	0.68	1.217
BERT+BiLSTM+CNN with SGD Optimizer without PSO	0.71	0.842
BERT+BiLSTM with SGD Optimizer without PSO	0.61	1.528
BERT+BiLSTM with PSO Optimizer	**0.81**	**0.974**
BERT+BiLSTM+CNN with PSO Optimizer	**0.84**	**0.543**

Table-2 – Comparative Evaluation on Mohler Dataset

VI. Conclusion

In this work, we have introduced a state of the art-new deep neural network models for the task of ASAG in Indian context. This article exposes the following theoretical implications through in-depth experimental comparison. Through the application of a combined model consisting of BERT, BiLSTM, and CNN, the functionality of ASAG is intended to be enhanced throughout the course of this research. We make use of two datasets, one of which is a new Computer science SAG dataset in Indian Context, and the other is a dataset compiled by Mohler. The suggested hybrid model is used for both training and testing of the datasets. The application of Particle Swarm Optimization enhances the outcomes. Following optimization, the accuracy for Computer Science SAG in Indian context dataset is 92%, while the accuracy for the Mohler dataset is 84%.

In the future, it is anticipated that it will be investigated whether or not adding an extra text corpus that is specific to a domain to a model that has already been pre-trained improves the model's capacity to comprehend language specific to that domain. Continued experimentation with strategies to reduce the amount of fine-tuning that is required (for example, by characterising the sorts of labelled samples that produce the biggest marginal improvement during fine-tuning, which enables more effective data collecting for automated grading). Finally, work on model maintenance, the reuse of models, and the development of efficient techniques to add new labelled samples to current fine-tuned methods will be of relevance so that a model can adapt over time. This can be accomplished by finding effective methods to add new labelled samples.

REFERENCES

[1] Michael Mohler and Rada Mihalcea. 2009. Text-to-Text Semantic Similarity for Automatic Short Answer Grading. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). Association for Computational Linguistics, Athens, Greece, 567–575.

[2] Gráinne Conole and Bill Warburton. A review of computer-assisted assessment. Research in learning technology, 13(1), 2005.

[3] Steven Burrows, Iryna Gurevych, and Benno Stein. The Eras and Trends of Automatic Short Answer Grading. International Journal of Artificial Intelligence in Education, 25(1):60–117, 2015.

[4] Keisuke Sakaguchi, Michael Heilman, and Nitin Madnani. Effective Feature Integration for Automated Short Answer Scoring. In NAACL-HLT, pages 1049–1054, 2015.

[5] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic Text Scoring Using Neural Networks. CoRR, abs/1606.04289, 2016.

[6] J. Kennedy and R. Eberhart, ``Particle swarm optimization," in Proc. Int. Conf. Neural Netw. (ICNN), vol. 4, 1995, pp. 1942_1948.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," pp. 4171–4186, 2018, arXiv.

[8] A. Vaswani et al., "Attention is all you need," in Proc. 31st Conf. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

[9] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing, 45 (11), pp. 2673–2681, 1997.

[10] S. Kumar, S. Chakrabarti, and S. Roy, "Earth movers distance pooling over Siamese LSTMs for automatic short answer grading," in Proc. Int. Joint Conf. Artif. Intell., 2017, pp. 2046–2052.

[11] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[12] B. Yang, L. Wang, D. F. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," in Proc. Conf. North Amer. Chapter Assoc. Computer. Linguistics, Hum. Lang. Technol., 2019, vol. 1, pp. 4040–4045.

[13] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, arXiv:1910.10683v3.

[14] J. Zhou, X. Huang, Q. Hu, and L. He, "SK-GCN: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification," Knowl.-Based Syst., vol. 205, no. 3, 2020, Art. no. 106292.

[15] W. Song, Z. Wen, Z. Xiao, and S. Park, "Semantics perception and refinement network for aspect-based sentiment analysis," Knowl. Based Syst., vol. 214, 2021, Art. no. 106755.

[16] W. X. Liao, B. Zeng, X. W. Yin, and P. F. Wei, "An improved aspect category sentiment analysis model for text sentiment analysis based on RoBERTa," Appl. Intell., vol. 51, pp. 3522–3533, 2021.

[17] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, arXiv:1907.11692v1.

[18] H. Yang, B. Zeng, J. Yang, Y. Song, and R. Xu, "A multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction," Neurocomputing, vol. 419, pp. 344–356, 2020.

[19] E. B. Page, "The imminence of grading essays by computers," Phi Delta Kappan, vol. 47, no. 5, pp. 238–243, 1966.

[20] Xinhua Zhu, Han Wu, and Lanfang Zhang, "Survey on Automated Short Answer Grading with Deep Learning: from Word Embeddings to Transformers," IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 15, NO. 3, JUNE 2022

[21] Ahmed Ezzat Magooda, Mohamed A. Zahran, Mohsen A. Rashwan, Hazem M. Raafat, and Magda B. Fayek. 2016. Vector Based Techniques for Short Answer Grading. In Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2016, Key Largo, Florida, USA, May 16-18, 2016, Zdravko Markov and Ingrid Russell (Eds.). AAAI Press, 238–243.

[22] Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In ICLR.

[23] Jeffrey Pennington, R. Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In EMNLP.

[24] Roy, S., Bhatt, H. S., & Narahari, Y. (2016). An iterative transfer learning-based ensemble technique for automatic short answer grading. arXiv preprint arXiv:1609.04909

[25] Zhu, X., Wu, H., & Zhang, L. (2022). Automatic Short-Answer Grading via BERT-Based Deep Neural Networks. IEEE Transactions on Learning Technologies, 15(3), 364-375.

[26] Condor, A., Litster, M., & Pardos, Z. (2021). Automatic Short Answer Grading with SBERT on Out-of-Sample Questions. International Educational Data Mining Society.

[27] Sung, C., Dhamecha, T. I., & Mukhi, N. (2019). Improving short answer grading using transformer-based pre-training. In Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20 (pp. 469-481). Springer International Publishing.