Automated essay scoring system using deep learning

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Engineering & Engineering/ Electronics & Communication Engineering/ Electrical & Electronics Engineering

By

19K41A04F5	B.Ramya
19K41A0594	B.Raju
19K41A0596	D.Sindhu Sri

Under the Guidance of
D. Ramesh

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S.R.ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA) Dec-2021.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled " Automated Essay Scoring using Deep Learning Algorithms" is a record of bonafide work carried out by the student(s) B.Ramya, B.Raju, D. Sindhu Sri, Roll No(s) 19K41A04F5, 19K41A0594, 19K41A0596 during the academic year 2021-22 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering/Electronics & Communication Engineering/Electrical & Electronics Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

In the previous years, various scholars have described the possible changes in grading assignments.Assessment in the Education system plays a crucial role in judging student performance. As the number of teachers' and student ratio is gradually increasing, the manual evaluation process becomes complicated. The drawback of manual evaluation is that it is time-consuming and heavy workload, lacks reliability, and many problems.Present Computer-based evaluation system works only for multiple-choice questions, but there is no proper evaluation system for grading essays and short answers.In recent times, ways and opportunities for giving feedback have changed as computer based programs have been more widely used in assessing students writing. Numerous researchers have studied computerized feedback and its potential.We studied the deep learning techniques used to evaluate automatic essay scoring and analyzed the limitations of the current studies and researches.

Keywords: AES, Automated essay scoring, Essay grading, features, Automatic features extraction,manual evaluation, Assessments.

.

Table of Contents

1. INTRODUCTION

Essays are considered as one of the very important evaluation criteria used by teachers to evaluate student's performance/Assessments.Manual Essay evaluation is a time consuming process, a teacher denotes a huge amount of time in evaluation of essays because of its subjectivity. Also because of subjective nature of the essay variation in grades usually occurs. Solution to such problem is automatic essay evaluation. Evaluating essays through computer will help reducing teachers load and time as well as reduce the variation in grades as a result of human factors.

The assessment plays a significant role in measuring the learning ability of the student. Most automated evaluation is available for multiple choice questions, but assessing short and essay answers remain a challenge. The education system is changing its shift to online-mode, like conducting computer-based exams and automatic evaluation. Here the problem is for a single question, we will get more responses from students with a different explanation. So, we need to evaluate all the answers concerning the question. Humans has various deviations so they are on not same mood every time sometimes they are happy or sad or angry etc it will effect on evaluation and grading of an essay.

It is a computer-based assessment system that automatically scores or grades the student responses by considering appropriate features.
Its objective is to classify a large set of textual entities into a small number of discrete categories, corresponding to the possible grades, for example, the numbers 1 to 6. Automated scoring means using machines to evaluate things typically evaluated by people.

It takes much less time to score, ensuring that results and feedback can be provided instantly. This is especially important with university and college classes that are so large it would be almost impossible to give each student frequent, detailed, individualized feedback. AES grades each essay based on its own merits, and similar papers will receive the same grade. It is beneficial for teachers they no need correct papers manually it decreases the burden. Students no longer need to wait for their score.

2. LITERATURE SURVEY

Essay writing is used for assessing academic achievements of a student which is an expensive, time-consuming. Manual grading of essays takes up substantial amount of instructor's time; therefore it's an expensive method. Human raters typically assess an essay's quality according to a scoring rubric that states the criteria an essay to meet a specific score level. Zhang (2013) argues that the quality of an essay is gauged by human raters with the help of a scoring rubric identifying the set characteristics. In some studies, instructors claim that they do not assign writing a research term paper longer than 5,000 words because it takes too long to mark papers.Zhang (2013) states, "Trained human raters are able to recognize and appreciate a writer's creativity and style (e.g., artistic, ironic, rhetorical), as well as evaluate the relevance of an essay's content to the prompt". According to Williamson at al. (2010), the process of essay scoring in which human effort is used is also time-consuming and thus if a considerable quantity is required to be scored, it becomes cumbersome. Utilizing technology can ease the burden human markers are under, however, student-instructor interaction can never be entirely replaced.

3. DESIGN:

1. REQUIREMENT SPECIFICATION(S/W & H/W)

Hardware Requirements

- **System** : Pentium 4, Intel Core i3, i5, i7 and 2GHz
- **RAM** Minimum : 4GB or above
- **Hard Disk** : 10GB or above
- **Input** : Keyboard and
- **Output** Mouse : Monitor or PC

Software Requirements

OS : Windows 8 or Higher

- **Platform** Versions : Jupiter
  Notebook/visual studio code
- **Program Language** : Python

3.2 Flow chart

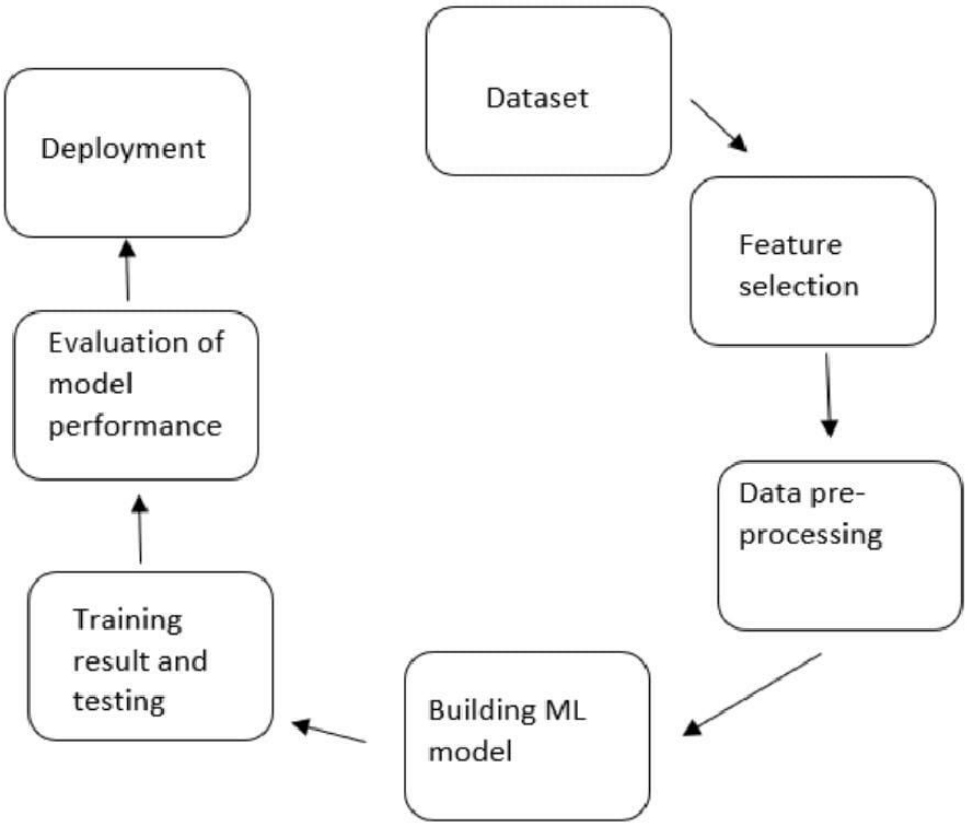

Figure 1 **-** flow chart

This is an analysis of the Automated essay grading system using the different deep learning model. For Analysis we use different deep learning tools like pandas, matplotlib, sci-kit-learn, etc.

4. DATASET:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	essay_id	essay_set	essay	rater1_do	rater2_do	rater3_do	domain1_	rater1_do	rater2_do	domain2_	rater1_tra	rater1_tra	rater1_tra	rater1_tra	rater1_tra	rater1_tra	rater2_tra	rater2_tra	rater2_tra	rater2_tra	rater2_tra	rater2_tra	rater3_tra
2	1	1	Dear local	4	4		8																
3	2	1	Dear @CA	5	4		9																
4	3	1	Dear, @CA	4	3		7																
5	4	1	Dear Local	5	5		10																
6	5	1	Dear @LO	4	4		8																
7	6	1	Dear @LO	4	4		8																
8	7	1	Did you kn	5	5		10																
9	8	1	@PERCEN	5	5		10																
10	9	1	Dear reade	4	5		9																
11	10	1	In the @LC	5	4		9																
12	11	1	Dear @LO	4	4		8																
13	12	1	Dear @CA	4	4		8																
14	13	1	Dear local	4	3		7																
15	14	1	My three c	3	3		6																
16	15	1	Dear, In th	3	3		6																
17	16	1	Dear @OR	6	6		12																
18	17	1	Dear Local	4	4		8																
19	18	1	Dear Local	4	4		8																
20	19	1	I aegre wa	2	2		4																
21	20	1	Well comp	3	3		6																
22	21	1	Dear @CA	4	4		8																
23	22	1	Dear local	2	1		3																
24	23	1	Dear local	5	5		10																
25	24	1	Dear local	6	5		11																
26	25	1	Dear @CA	4	4		8																
27	26	1	Do you thi	5	4		9																
28	27	1	Computers	2	2		4																
29	28	1	Dear News	5	4		9																
30	29	1	Dear local	5	4		9																

Figure 2 :dataset

5. DATA PREPROCESSING:

Dataset is a collection of data. For Automated essay grading we need dataset. The dataset to work on and we decided to use the Long short term memory Model. Create a model that will help us to estimate of score for given essay by considering features like essay Id, essay set, essay and we get output as score of the given essay.Dataset consists of 12979 rows and 28 coloums.

Essay set	Number Number of an essay set{1,2,3….}
Essay	String Large text data ,cillection of paragraphs.
Essay id	Number Number {1,2,3….}
Score	Number Number {1,2,3….}

Table 3. Attributes Validation

6. METHODOLOGY:

This section talks about the algorithms used for the project. We used LSTM(Long Short Term Memory).

LSTM(LONG SHORT TERM MEMORY)

deep learning model based LSTM(Long short term memory) method for grading essays automatically

During computation of long essays which are far away,it is impossible to store which causes vanishing of gradient. In order to maintain we use LSTM(Long Short-Term Memory Network.

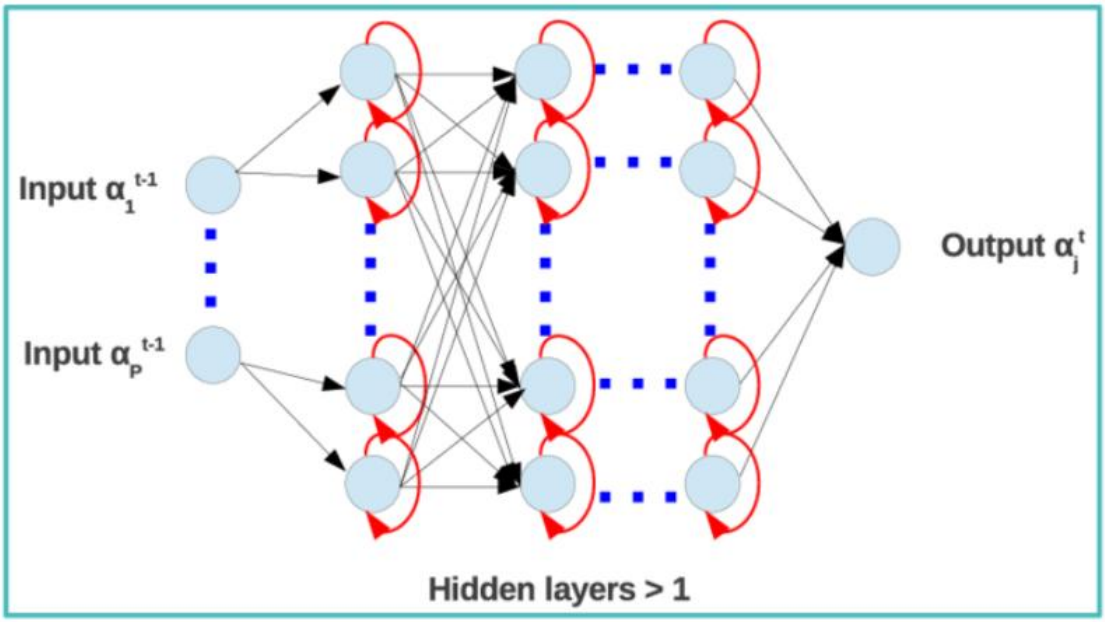

Fig 4:LSTM Model

LSTMs have three types of gates they are:

Input gates, forget gates, and output gates which controls the flow of information. The hidden layer output of LSTM includes the hidden state and the memory cell. Only the hidden state is passed into the output layer. The memory cell is entirely internal.

This challenge to address long-term information preservation and short-term input skipping in latent variable models has existed for such a long time. One of the earliest approaches to address this was the long short-term memory (LSTM) . It shares many of the properties of the GRU. Interestingly, LSTMs have a slightly more complex design than GRUs but predates GRUs by almost two decades.

To control the memory cell we need a number of gates. . We will refer the *output gate* to read out the entries from the cell. *input gate a* second gate is needed to decide when to read data into the cell. Last, we need a mechanism to reset the content of the cell, governed by a *forget gate*. The motivation for such a design is the same as that of GRUs, namely to be able to decide when to remember and when to ignore inputs in the hidden state via a dedicated mechanism

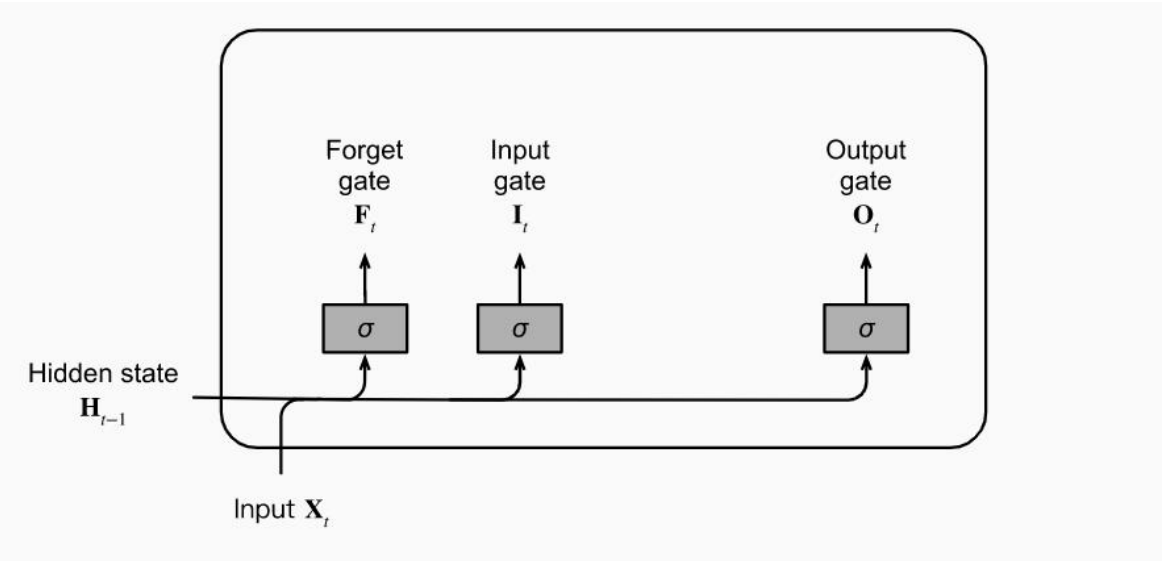

Fig 5:gated memory cell in lstm

RNN will estimate word/character/number the sequence based on previous time stamps not based on future dependency

- Due to deep structure, RNN has vanishing gradient problem
- Due to deep structure, RNN may has exploding gradient problem
- Due to deep structure, the value at particular time stamp may forget initial values in RNN.

So, to overcome this problem we used LSTM which solves the vanishing gradient problem in RNN.

Vanishing gradient:
The vanishing gradients is a problem and an example of the unstable behaviour of a multiple layered neural network. Neural networks are unable to back propagate the gradient information to the input layers of the model.

Output at particular time stamp highly depends on near time stamps and weakly effected by initial time stamps. Gradient of error will be have neglisible impact on initial weights (Gradient vanishing). Some times gradients may increase to high value (Exploding gradient), will lead high values of parameters. Exploding gradient can be overcome by using gradient clipping method.

The gradients keep getting smaller and smaller when moving from the top layer to the lower layers. Due to this, sometimes the weights in the lower layers do not get updated. For that reason, the training of the network is not very effective. Ans this is known as the vanishing gradients problem.

It is not very uncommon for deep neural networks to suffer from the vanishing gradients problems. One of the reasons for this is the use of the logistic sigmoid activation function.

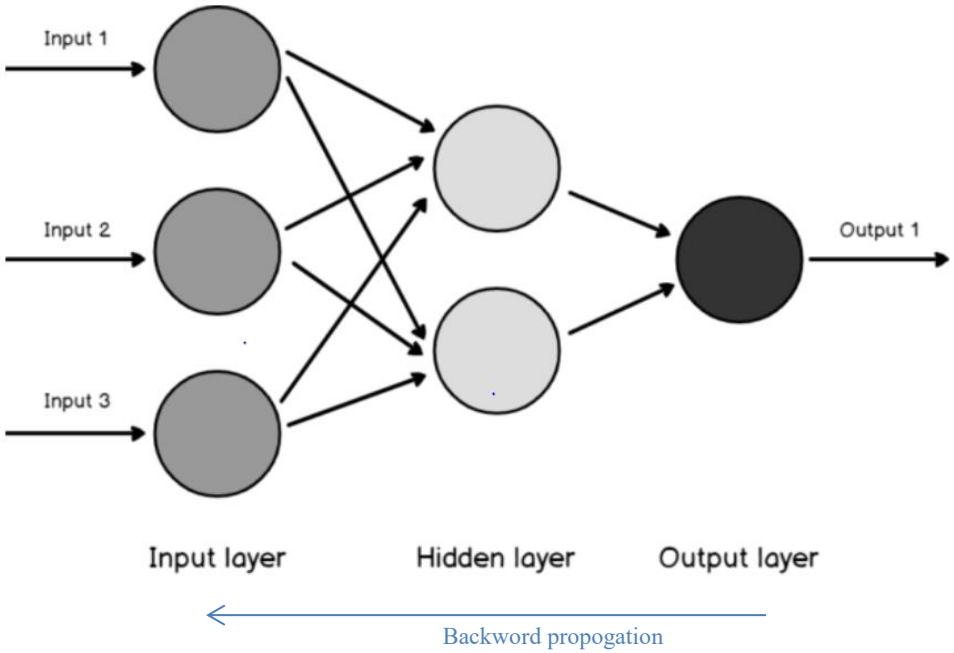

Fig 6:propogation between layers

ReLU (rectified linear activation function):

ReLU -The rectified linear activation function or ReLU for short is a piece wise linear function that will give output the input value directly if it is positive value, otherwise, it will output zero. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time.

$$f(u) = \max(0, u)$$

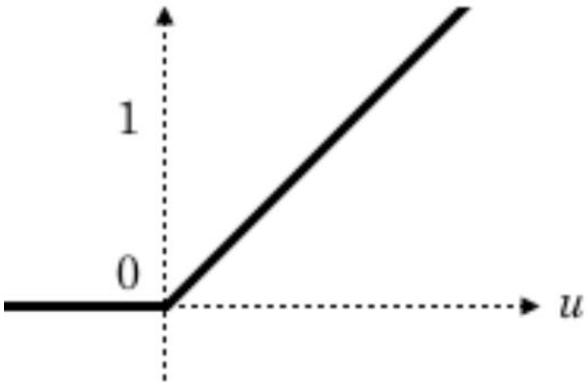

Fig 7:ReLU activation function graph

ReLU solve vanishing gradient problem:

ReLU can Solve the Vanishing Gradient Problem.
If the input value is positive, the ReLU function returns it; if it is negative, it returns 0. The ReLU's derivative is 1 for values larger than zero. Because multiplying 1 by itself several times still gives 1, this basically addresses the vanishing gradient problem.

7.RESULTS:

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_16 (LSTM)               (None, 1, 300)            721200

 lstm_17 (LSTM)               (None, 64)                93440

 dropout_8 (Dropout)          (None, 64)                0

 dense_8 (Dense)              (None, 1)                 65

=================================================================
Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0
_________________________________________________________________
Epoch 1/6
163/163 [==============================] - 7s 25ms/step - loss: 65.2524 - mae: 4.4162
Epoch 2/6
163/163 [==============================] - 4s 27ms/step - loss: 41.8929 - mae: 3.6821
Epoch 3/6
163/163 [==============================] - 5s 28ms/step - loss: 34.3603 - mae: 3.4905
Epoch 4/6
163/163 [==============================] - 5s 28ms/step - loss: 31.8813 - mae: 3.4519
Epoch 5/6
163/163 [==============================] - 5s 28ms/step - loss: 29.7178 - mae: 3.2908
Epoch 6/6
163/163 [==============================] - 5s 28ms/step - loss: 27.9376 - mae: 3.0976
Kappa Score: 0.8392147400281298
```

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 lstm_18 (LSTM)               (None, 1, 300)            721200

 lstm_19 (LSTM)               (None, 64)                93440

 dropout_9 (Dropout)          (None, 64)                0

 dense_9 (Dense)              (None, 1)                 65

=================================================================
Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0
_________________________________________________________________
Epoch 1/6
163/163 [==============================] - 9s 25ms/step - loss: 65.3300 - mae: 4.3739
Epoch 2/6
163/163 [==============================] - 5s 28ms/step - loss: 40.6654 - mae: 3.6014
Epoch 3/6
163/163 [==============================] - 5s 28ms/step - loss: 34.0423 - mae: 3.4777
Epoch 4/6
163/163 [==============================] - 4s 23ms/step - loss: 30.6998 - mae: 3.3739
Epoch 5/6
163/163 [==============================] - 4s 26ms/step - loss: 27.9688 - mae: 3.2043
Epoch 6/6
163/163 [==============================] - 4s 23ms/step - loss: 27.0539 - mae: 3.0428
Kappa Score: 0.8393407306863854
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_20 (LSTM)              (None, 1, 300)            721200

 lstm_21 (LSTM)              (None, 64)                93440

 dropout_10 (Dropout)        (None, 64)                0

 dense_10 (Dense)            (None, 1)                 65

=================================================================
Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0
_________________________________________________________________
Epoch 1/6
163/163 [==============================] - 7s 22ms/step - loss: 61.7039 - mae: 4.2946
Epoch 2/6
163/163 [==============================] - 4s 22ms/step - loss: 39.7976 - mae: 3.5957
Epoch 3/6
163/163 [==============================] - 4s 22ms/step - loss: 33.0700 - mae: 3.4526
Epoch 4/6
163/163 [==============================] - 4s 24ms/step - loss: 30.4673 - mae: 3.3701
Epoch 5/6
163/163 [==============================] - 4s 25ms/step - loss: 28.6825 - mae: 3.2062
Epoch 6/6
163/163 [==============================] - 4s 26ms/step - loss: 26.4508 - mae: 3.0264
Kappa Score: 0.8437863101065961
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_22 (LSTM)              (None, 1, 300)            721200

 lstm_23 (LSTM)              (None, 64)                93440

 dropout_11 (Dropout)        (None, 64)                0

 dense_11 (Dense)            (None, 1)                 65

=================================================================
Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0
_________________________________________________________________
Epoch 1/6
163/163 [==============================] - 9s 35ms/step - loss: 63.6331 - mae: 4.3034
Epoch 2/6
163/163 [==============================] - 6s 37ms/step - loss: 39.8944 - mae: 3.5979
Epoch 3/6
163/163 [==============================] - 6s 36ms/step - loss: 33.4126 - mae: 3.4933
Epoch 4/6
163/163 [==============================] - 5s 33ms/step - loss: 30.3313 - mae: 3.4064
Epoch 5/6
163/163 [==============================] - 5s 34ms/step - loss: 28.1997 - mae: 3.2350
Epoch 6/6
163/163 [==============================] - 6s 34ms/step - loss: 26.5343 - mae: 3.0367
Kappa Score: 0.8107767009961792
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_24 (LSTM)              (None, 1, 300)            721200

 lstm_25 (LSTM)              (None, 64)                93440

 dropout_12 (Dropout)        (None, 64)                0

 dense_12 (Dense)            (None, 1)                 65

=================================================================
Total params: 814,705
Trainable params: 814,705
Non-trainable params: 0
_________________________________________________________________
Epoch 1/6
163/163 [==============================] - 8s 25ms/step - loss: 63.8552 - mae: 4.3827
Epoch 2/6
163/163 [==============================] - 4s 26ms/step - loss: 39.9030 - mae: 3.5821
Epoch 3/6
163/163 [==============================] - 5s 28ms/step - loss: 33.5887 - mae: 3.4873
Epoch 4/6
163/163 [==============================] - 5s 29ms/step - loss: 30.8679 - mae: 3.3910
Epoch 5/6
163/163 [==============================] - 4s 28ms/step - loss: 28.3316 - mae: 3.2413
Epoch 6/6
163/163 [==============================] - 5s 28ms/step - loss: 26.6824 - mae: 3.0573
Kappa Score: 0.8371404898235586
```

```
Average Kappa score after a 5-fold cross validation:  0.8341
```

Figure 8. Result

RESULT

NEURAL NETWORK LOSS AND MAE

NUMBER OF FOLDS		NUMBER OF EPOCHS 1	2	3	4	5	6	kappa score
1	LOSS	65.2	41.8	34.3	31.8	29.7	27.9	0.83
	MAE	4.4	3.6	3.4	3.4	3.2	3.0	
2	LOSS	65.3	40.6	34.0	30.6	27.9	27.0	0.83
	MAE	4.3	3.6	3.4	3.3	3.2	3.0	
3	LOSS	61.7	39.7	33.0	30.4	28.6	26.4	0.84
	MAE	4.2	3.5	3.4	3.3	3.2	3.0	
4	LOSS	63.6	39.8	33.4	30.3	28.1	26.5	0.81
	MAE	4.3	3.5	3.4	3.4	3.2	3.0	
5	LOSS	63.8	39.9	33.5	30.8	28.33	26.68	0.83
	MAE	4.3	3.5	3.4	3.3	3.2	3.0	

Average kappa score after 5-fold cross validation :0.834

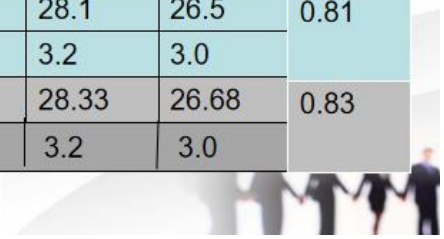

Fig 9:final result

The neural network deep learning algorithms that we used is LSTM(long-short term memory). Thisalgorithms worked well on Automated essay geading system. We gor average Kappa score as 83% for four foulds. LSTM has four layers 1)LSTM layer-1 2)LSTM layer-2 3)drop out layer 4)dense layer.

8. CONCLUSION:

The previous or existing automated essay grading systems used traditional text based machine learning models. The results highly rely on the crafted extracted features. The performances are unstable when grading different essays from various topics.

So,we propose a deep learning model based LSTM(Long short term memory) method for grading essays automatically. The neural network deep learning algorithms that we used is LSTM(long-short term memory). Thisalgorithms worked well on Automated essay geading system. We gor average Kappa score as 83% for four foulds.

During computation of long essays which are far away,it is impossible to store which causes vanishing of gradient. In order to maintain we use LSTM(Long Short-Term Memory Network)

9. REFERENCES:

1. https://www.researchgate.net/publication/353639974_A_Comprehensive_Review_of_Automated_Essay_Scoring_AES_Research_and_Development
2. https://www.frontiersin.org/articles/10.3389/feduc.2020.572367/full
3. https://www.google.com/url?sa=t&source=web&rct=j&url=https://files.eric.ed.gov/fulltext/EJ843855.pdf&ved=2ahUKEwjVqMqM1Kz3AhUTxzgGHQ8yBTIQFnoECEkQAQ&usg=AOvVaw1W-udAeEeEucePqr3e-nR4
4. https://link.springer.com/article/10.1007/s10462-021-10068-2
5. https://analyticsindiamag.com/a-complete-guide-to-lstm-architecture-and-its-use-in-text-classification/
6. https://www.mdpi.com/2073-8994/10/12/682/pdf
7. https://www.ijrte.org/wp-content/uploads/papers/v8i6/F9938038620.pdf
8. https://www.academia.edu/Documents/in/Automated_Essay_Scoring
9. https://link.springer.com/article/10.1007/s41237-021-00142-y