Automated Essay Scoring System with Grammar Score Analysis

Aniket Ajit Tambe
Computer Engineering
Veermata Jijabai Technological Institute
Mumbai, India
aniketajittambe@gmail.com

Manasi Kulkarni
Computer Engineering
Veermata Jijabai Technological Institute
Mumbai, India
mukulkarni@ce.vjti.ac.in

***Abstract*—An Automated Essay Scoring System is a system that deals with grading Hand written essays without any human intervention. Most of the research done in this field involves direct mapping of an essay's numerical representation, using word embedding, to its golden score, without any specific trait scoring. So, this research aims to use latest contextual Text Embedding i.e., BERT Embedding for numerical representation of Essay and granulate the scoring of essay into two modules: structure scoring module which deals with scoring essay on the basis of its structure and grammar scoring module, which deals with scoring essay on the basis of its grammatical correctness. To evaluate the proposed model, Quadratic Weighted Kappa Score is used. In this implementation, a QWK score of 0.75 for Structure score and 0.70 for Grammar Score has been obtained. This research and its specific trait scoring can be used as base for future implementation with more detailed feature specific scoring tasks and improve the scope of grammar scoring by considering more grammatical cases.**

***Index Terms*—Automated Essay Scoring, BERT, Quadratic Weighted Kappa, Corpus of Linguistic Acceptability, Regression , Sequential Model.**

I. Introduction

An Essay is a short form formal piece of writing that deals with a single subject line, mostly used in the field of education to convey student's understanding about a specific topic. Essays can be graded on the basis of following factors:

- Type of Vocabulary used
- Structure of essay
- Grammatical Error
- Constraint errors (like spelling mistakes, word count, sentence count etc.)

While the type of vocabulary used, structure accounts for the structure aspect of the essay, grammar and constraint like spelling mistakes account for the Grammar aspect of the essay. Grading multiple such essays manually and repeatedly can result in fatigue and unwanted biases which can affect the scoring and make it inconsistent. To replace the manual task of scoring and automate the task, an Automated Essay Scoring system is being looked into, which harnesses the benefits of NLP and Machine Learning. One important factor to consider while developing an automated essay scoring system is to use numerical representation of the essay, such that the context of words and sentences used in the essay still hold. One such example is the BERT Embedding. Another important factor to look into while grading essays, which hasn't been looked into more, is grading essays on the basis of Grammar. Hence, the goal of this research is to implement Automated Essay Scoring System that grades essay without any human intervention, uses Latest Context embedding i.e., BERT Text Representation for numerical representation of essay and enhances AES system's functionality with Grammar Error Detection. For developing such a model, the ASAP AES Data Set hosted by Hewlett Foundation on Kaggle[1] can be used to train, validate
, test the model.

II. Related Work

To form a base knowledge about the proposed system, several previous implementations were looked into. Starting with the oldest approach[2], which used Recurrent neural Network trained on Regression without any specific Feature Engineering. Along the different layers of RNN, the model automatically selects a feature and uses it to do the final task of predicting overall score. The numerical representation used in this approach did not consider context of essay's constituent parts. This approach was able to clock a QWK score of 0.72. Another approach used for AES used the new technology of BERT a.k.a Bidirectional Encoder Representations from Transformers released by Google[3] which was specifically used because of its bidirectional nature which holds context. To show BERT's advantage in implementation, in another research[4] author implemented three AES system using three different type of representation used are:

- Glove (Global vectors for word representation)
- ELMo (Embedding from Language Models)
- BERT (Bidirectional Encoder Representations from Transformers

While Glove is word embedding which does not hold context, BERT and Elmo and context embedding which hold context difference being Bidirectional nature and concatenating two opposite direction approaches, respectively. After comparing the performance of all the three above embedding using QWK Score, it was concluded that BERT gave better results. This can be iterated upon to make the system more granular when it comes to specific tasks. Another research[5] dwells on the answer for the question whether Fine tuning BERT model is worth it in the field of AES. It was observed that fine tuning

978-1-6654-6047-7/22/$31.00 ©2022 IEEE

BERT model did give better QWK and was preferred over Pre-training. To look into the Grammar error detection part, while no specific research has been done to implement Grammar error detection in AES, independent research has been done to check grammatical acceptability of individual sentences. One such research[6] involves developing a neural network model which uses the Corpus of Linguistic Acceptability data set for training. Once trained, this model is able to predict the Grammatical Acceptability of Sentence with an Matthew Correlation Coefficient Score[7] of 0.52. The model is able to predict the correctness of individual sentences for certain types such as Syntactical correctness, Local Semantic Correctness, Morphological Correctness. Even though this model fails to work on other cases of sentences, as a pioneer approach for implementing in AES, this work can be iterated upon to work in the case of AES at individual sentence level.

III. System Design

The goal of this research is to implement proposed solution for improved AES, targeting below objective:

1) BERT Text representation to hold the context of each word.
2) Attempt to implement Grammar Error detection by checking grammar of each sentence of the essay.

To encapsulate these objectives, we have developed the system architecture depicted in 1: The proposed architecture consists

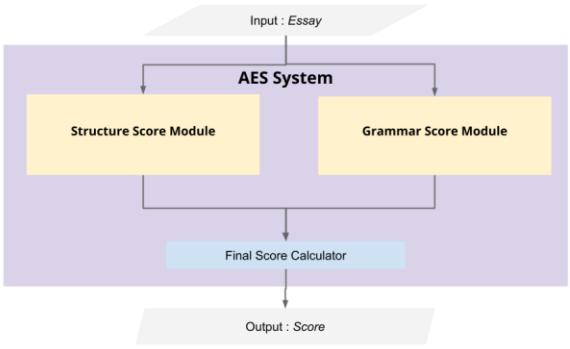


Fig. 1. Proposed System Architecture.

of two modules. They are:

- Structure Score Module
- Grammar Score Module

Same input will be provided to these parallel modules, where they will be preprocessed and worked upon according to the requirement of core tasks carried out in each module.

A. Essay Structure Scoring Module

Essay Structure Scoring module will deal with scoring the structure part of the essay. That is, scoring is done on the basis of traits like organization, type of vocabulary, combination of words used etc. For this BERT text representation [8] is used where each essay in the resultant vector has 768 dimensions and then using this vector representation of the essay a trained model is used to predict score. Below is the activity flow for

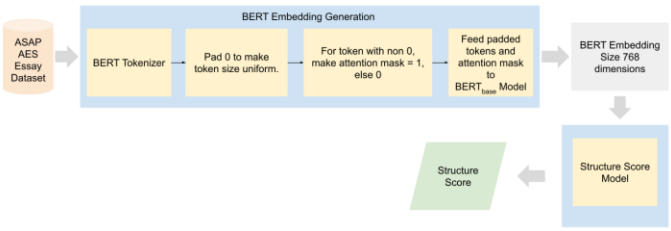


Fig. 2. Testing Essay Structure Scoring.
Testing Essay Structure Scoring.

TABLE I
Structure Score Model Summary

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional)	(None, 1, 800)	3740800
bidirectional_1 (Bidirectional)	(None, 256)	951296
dropout (Dropout)	(None, 256)	0
dense (Dense)	(None, 1)	257
Total params: 4,692,353		
Trainable params: 4,692,353		
Non-trainable params: 0		

the structure scoring task: As shown in the Fig **??** above, essay goes through below layers:

1) **Essay to BERT Text Embedding:** Here essay is tokenized to be compatible to generate BERT Embedding. Along with BERT Tokens, an attention mask is created with 1 and 0 where 1 means token and 0 means padding. This tokenized essay and its attention mask is fed to the BERT base [2] model which generates a vector representation of size 768 dimension, which holds the numerical representation of our essay.
2) **Structure Score Model:** Once the vectors are generated, it is fed to the Structure Score Model. This model is developed using a Sequential Model which consists of two LSTM Layers (dimension 400 and 128), dropout layer with dropout set at 0.5 and Dense layer using Relu activation function[9] and one output label for score. Table I shows the model summary for the developed structure score model.

This module will score the essay on the basis of its structure component such as the type of vocabulary and combination of vocabulary used.

B. Grammar Scoring Module

The second parallel module, Grammar Scoring module deals with finding the grammar error in each sentence of the essay. To be precise, this module generates several features from the essay which hold grammatical scoring information and levies them to calculate grammar score. For this a grammar checker model has been developed, iterating on previous research of Neural acceptability of sentence grammar[6] [2].

1) **Grammar Checker Model:** In this a model has been developed by fine tuning BertSequenceClassification for binary task i.e. If a given sentence is grammatically correct or not. For fine tuning, we have used the CoLA data set[8] which consists of individual sentences and each sentence masked as grammatically correct(1) or not(0). Fig 3 represents the flow diagram for grammar checker model:

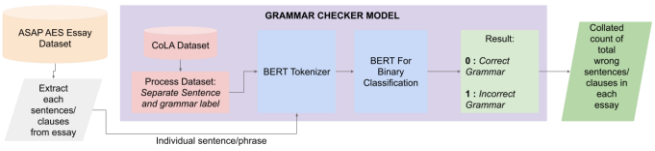


Fig. 3. Training Grammar Checking Model.

For developing the grammar checker model, we have used CoLA Data set internally. The individual sentence of this data set is tokenized using BERT Tokenizer and attention mask is created, which is used to fine tune BERT Sequence classification model. Once this is done, the model is ready to predict whether a sentence is grammatically acceptable or not. This grammar checker model will be used as one of the components inside grammar scoring module to predict the score by generating several features.

2) **Feature Creation for Grammar Score Calculation :** The important feature generated by the grammar score module from each individual essay can be seen in the Fig 4. Several outside libraries are used for this feature generations. Such as to fix the spelling of words, two libraries were used, TextBlob Library[10] and JamSpell library[11]. These libraries are used to generate features such as total sentence ,clauses , words ; grammatically wrong sentences/clauses before and after fixing spelling mistakes. Also, to extract clauses from sentences, SpaCy's dependency parser [12] is used.

3) **Score Mapping Function:** Here using the above generated numerical features, Logistic regression is used to train the model to predict the golden score.

IV. Experimental Setup

A. Data set

Two data sets have been used.

1) ASAP AES Data Set: ASAP AES Data Set[1] hosted by Hewlett Foundation on Kaggle site is used as primary data set. This data set consists of 8 Sets of Essays out of which 7Th set is used as it has separate grammar score which is imperative for the proposed system. In this set, the essays consist of an average 11 lines and 250 words per essay. There are a total 1569 essays in this set. In this set, essays are score on following traits by two raters, with their marks range mentioned ahead of them:

- Trait 1: Ideas (0-3)
- Trait 2: Organization (0-3)
- Trait 3: Style (0-3)
- Trait 4: Grammar (0-3)

For proposed system's implementation, the said data set is preprocessed according to following steps:

1) Combine both rater's trait 1,2,3 marks to get StructureScore.
2) Combine both rater's trait 4 marks to get Grammar Score.
3) All other features except essay id, essay set no, essay, Structure score, grammar score; drop rest.
4) Both structure score and essay score are normalized to be out of 10.
5) Data Set is ready with required features.

Fig 5 is a snippet of the actual data set after pre-processing, that will be used in both the module: structure score and grammar score.

This preprocessed data set of 1568 essays is further divided into two as shown in Fig 6. One portion containing 1000 essays for training and testing and evaluation. Another portion 568 essays for analysis on completely blind new set.

2) CoLA Data Set: CoLA Data set or Corpus of Linguistic Acceptability Data set[13] specifically used to train the Grammar Checking Model of our Grammar Score Module. It is not part of the AES system's essay. This data set consist of set of 10000 sentences and each sentence containing label in front of it in the form of 0 and 1, where:

- 0 means grammatically wrong sentence
- 1 means grammatically correct sentences.

Fig 7 is a snippet of the data set record, along with features that are specifically used in our implementation, highlighted. This data set, after preprocessing, is fed to our BERT Binary Classification Model for fine tuning the BERT Model to predict the grammatical acceptability of a sentence.

B. Evaluation Metric

1) Quadratic Weighted Kappa, QWK: For evaluation Quadratic Weighted Kappa Score (QWK)[14] is used for each module. This score deals with the agreement between model rating and human rating mentioned in the data set. That is , it gives a cumulative level of agreement between the AES model's score and the golden score mentioned in the data set . The score can vary from -1 to +1., where:

- -1 means complete disagreement
- 0 means random agreement
- +1 means complete agreement

The QWK Score can be calculated using the following formula, once all scores are obtained:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

$$k = 1 - \frac{\sum W \cdot O}{\sum W \cdot E}$$

Where,

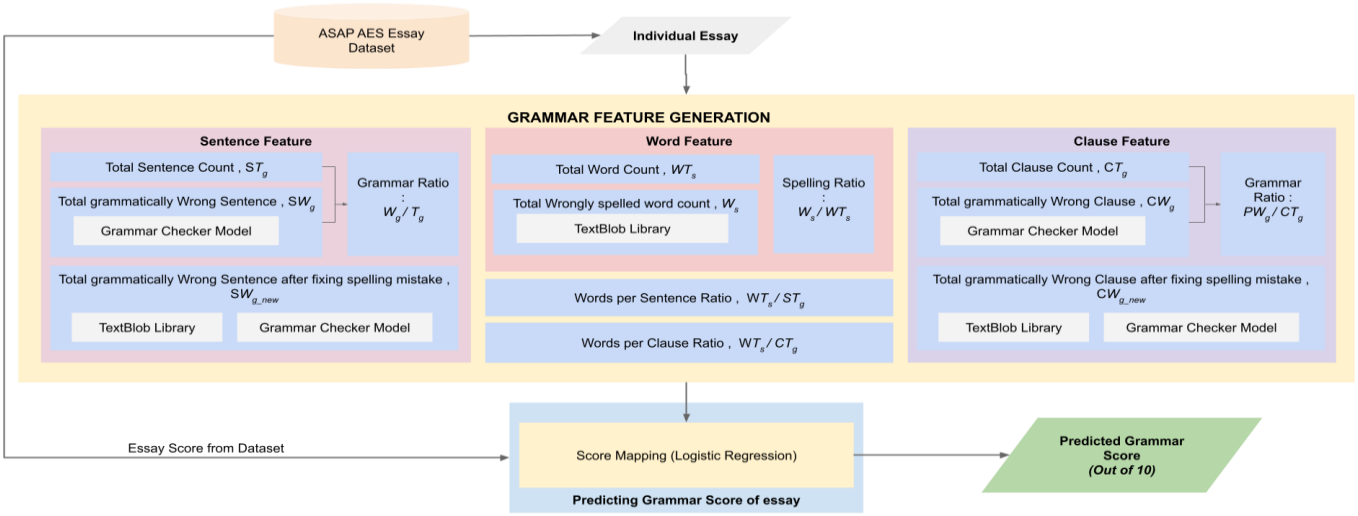


Fig. 4. Grammar Scoring Module Features Generated.

	essay_id	essay_set	essay	structure_score	grammar_score
0	17834	7	Patience is when your waiting .I was patience ...	5	8
1	17836	7	I am not a patience person, like I can't sit i...	5	5
2	17837	7	One day I was at basketball practice and I was...	6	6

Fig. 5. ASAP AES Set 7 Data Set's record after pre-processing.

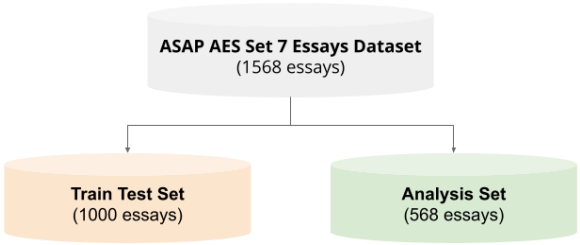


Fig. 6. ASAP AES Data set train-test and Analysis Set split

	sentence_source	label	label_notes	sentence
0	gj04	1	NaN	Our friends won't buy this analysis, let alone...
1	gj04	1	NaN	One more pseudo generalization and I'm giving up.
2	gj04	1	NaN	One more pseudo generalization or I'm giving up.

Required Attributes

Fig. 7. CoLA Data Set record.

- W is the weight matrix which specifies the penalty value between human rater i and AES system rater j.
- N is the number of possible score labels.
- O is Confusion matrix where Oi,j is the total number of essays that receive a score label i in human rater and j in AES system.
- E is the Expected value matrix where Ei,j being the product of the number of essays that received a score label i by the human raters and score j by the AES.

C. Training

To train our individual modules, first of all the ASAP AES's data set is obtained and pre-processed as said before, i.e. Merge trait scores into Structure and Grammar Score and Normalize the derived score to be out of 10 a. Once that is done, each of the proposed modules will take the processed data set as input.

1) Structure Scoring Training: For essay structure scoring module, The Processed ASAP AES data set's essay and structure acts as the sole input to our model. The essay is tokenized to be BERT compatible using BERT Tokenizer and also an attention mask is created. This Tokenized input, attention mask, and structure score are used as input to train our essay structure scoring module over several epochs, to predict the structure score.

2) Grammar Scoring Training: For Grammar Scoring, another data set along with processed ASAP AES data set is used as input. The CoLA data set is used to train the grammar checker model. For this the individual sentence and its grammar labels are used. The sentence is BERT Tokenized and Attention mask is created. The generated token and mask along with label is used to fine tune the BERT Sequence classification model to cover the task of Binary classification to predict whether a sentence is grammatically acceptable or not. Once this model is trained, the original data set i.e., ASAP AES data set's essay and grammar score is used for our module. In this each sentence of essay is separated, BERT Tokenized and masked, then it is passed to grammar checker model to extract the count of wrong sentence per essay. Along with this more features such as total words, sentences, wrongly spelled words, grammar error count after fixing spelling mistake, words per sentence are generated. These numerical features act as input to train our Grammar score module using Regression to predict the grammar score by using a train test split of 70-30.

V. Results and Analysis

A. Results

Below are the details for each module's result

1) Structure Score Module: The Structure Module part of our system was able to clock the best QWK score of 0.75 when the epoch was set at 70. On the analysis set, it was observed that the structure score module was able to churn out a QWK score of 0.76.

2) Grammar Score Module: Similarly in the Grammar Score Module, after training the final Regression model using the said feature, the model was able to clock a QWK score of 0.65 during training testing . On the analysis set, it was observed that the grammar score module was able to churn out a QWK score of 0.68.

B. Analysis

The trained model's performance was checked on the analysis set which consisted of 568 essays to get an unbiased performance and evaluation. Below are the findings of it.

1) Structure Score Module: On the Analysis set, the model was able to crunch out a QWK score of 0.76 . When the predicted structure score was compared with the golden structure score, in the form of a line plot graph, Fig 8 results were visible.

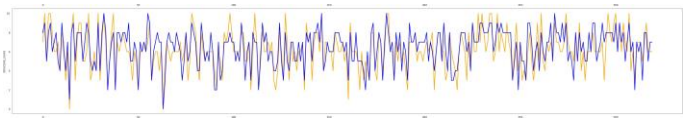


Fig. 8. Golden Structure Score (orange annotated) vs Predicted Structure Score (blue annotated)

The graph pretty much coincides when plotted together where orange annotated line represent the golden score and blue annotated line represent the predicted score. This visualizes the accuracy of the structure score module.

2) Grammar Score Module: Similarly for Grammar score was able to top out at 0.68 QWK on analysis set. When the golden and the predicted grammar score was plotted on the line plot as shown in Fig 9, it was observed that for certain peaks and troughs, the lines weren't completely coinciding. This is due to the 0.68 score which was brought in due to less accuracy of relatively new implementation of grammar checker.

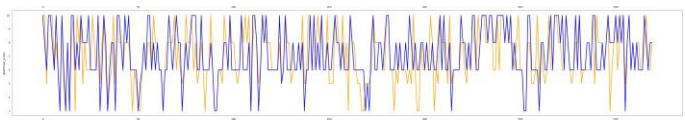


Fig. 9. Golden Grammar Score (orange annotated) vs Predicted Grammar Score (blue annotated)

3) Total Automated Essay Scoring System: When the com plete AES model's performance was evaluated on the Analysis Set, it was observed that a QWK score of 0.78 was obtained which was comparatively good with respect to previous implementation, considering two separated module for individual trait score calculation was implemented. The line plot of total golden and predicted score showed a fair share of coincidence except at few places which can be accounted to the low score of grammar score module as seen in Fig 10.

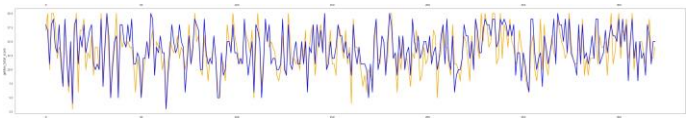


Fig. 10. Golden Total Score (orange annotated) vs Predicted Total Score (blue annotated)

From Table II, we can see that when structure score's difference between golden and predicted sore was plotted , majority of data points lied in the score difference of 0 and 1. Which is comparatively good. But when same was plotted for Grammar Score, there was a sudden spike in the score difference of 2 too. This was due to performance of grammar checker model, which due to being a first implementation, has started low but can be improved in future work. When the total score difference was plotted, which happens to be sum of both structure and grammar score, the impact of grammar score's performance can be seen due to presence of bar representing the score difference of 2.

C. Selecting Different Features for Grammar Score

Once the model for both structure and grammar score were developed and different features were generated, the QWK score was observed for both test set and analysis set. A pre build library called GramFormer(GF)[15] was also used for comparison purpose to see the developed model ,i.e. Grammar Checker Model(GCM) performs with same set of features when compared with GramFormer library. From the results obtained, the best set of features along with our developed model was selected for final implementation. The result for such different combination of features can be seen in Table III. The best feature combination is highlighted in green, which happens to be Sentence and Clause extraction and checking its grammatical acceptability using the developed Grammar Checker Model.

D. Individual Feature vs Score

Now several features were considered when calculating the score for each essay. Some of the important features that were considered were the total sentence/clause, total grammatically wrong sentences/clause count, words per sentence/clause. Below Analysis shows how much each individual feature, represented on Y axis contributed in for both golden and predicted score, represented on X axis, when taken average per score label. Here on X axis, blue represents the predicted score and orange represents the golden score and Y represent individual features considered for comparison.

TABLE II
GOLDEN VS PREDICTED SCORE DIFFERENCE FOR STRUCTURE, GRAMMAR AND TOTAL SCORE

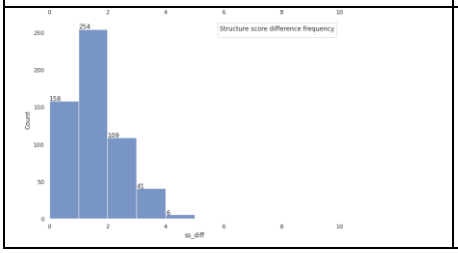
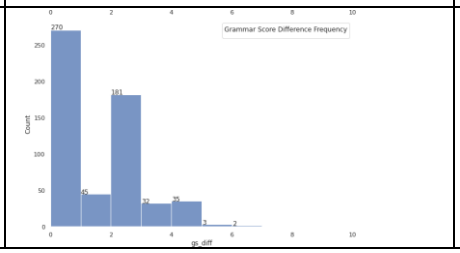
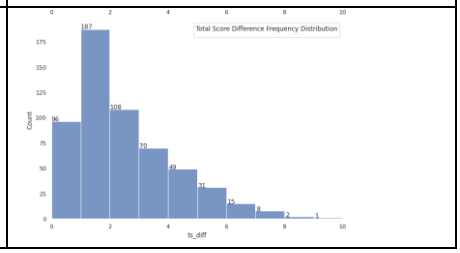
Structure Score Difference	Grammar Score Difference	Total Score Difference
		

TABLE III
QWK SCORE FOR DIFFERENT FEATURE COMBINATION

Version no	**Features Considered**	**QWK Score on Test Set**	**QWK Score on Analysis Set**
v1	Sentence Features (GCM) + Word Features	0.66	0.64
v2	Sentence Features (GCM) + Clause Features (GF)	**0.71**	**0.64**
v3	Sentence Features (GCM) + Clause Features (GCM)	**0.73**	**0.68**
v4	Sentence Features (GCM) + Clause Features (GCM) + Word Features	0.70	0.60
v5	Sentence Features (GF) + Clause Features (GF)	0.73	0.64
v6	Sentence Features (GCM) + Clause Features (GCM)+more sentence and Clause ratio	0.71	0.65
v7	Sentence Features (GCM) + Clause Features (GCM)+sentence and Clause ratio + Word Features	0.73	0.60

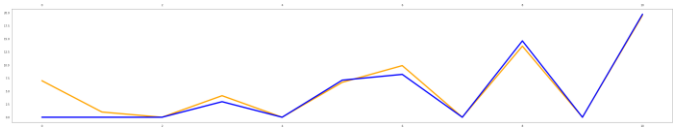


Fig. 11. Total Sentence vs Golden (orange annotated)/Predicted (blue annotated) Grammar Score

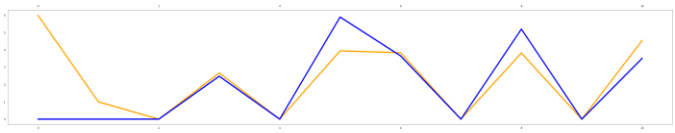


Fig. 12. Total Grammatically wrong sentences vs Golden (orange annotated)/Predicted(blue annotated) Grammar Score

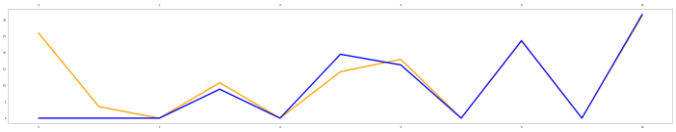


Fig. 13. Total Clause vs Golden (orange annotated)/Predicted (blue annotated) Grammar Score

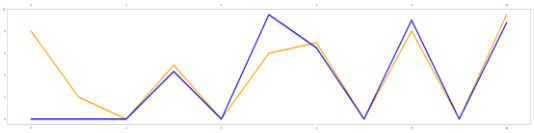


Fig.14. Grammatically wrong clauses vs Golden (orangeannotated)/Predicted (blue annotated) Grammar Score

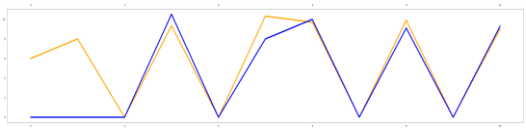


Fig. 15. Total words per clause vs Golden (orange annotated)/Predicted (blueannotated) Grammar Score

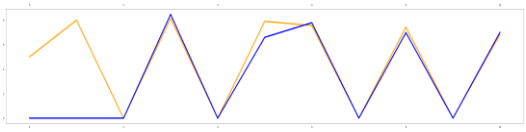


Fig. 16. Total words per sentence vs Golden (orange annotated)/Predicted (blueannotated) Grammar Score

From the Individual feature vs Score analysis depicted in Fig 11 , Fig 12 , Fig 13, Fig 14 , Fig 15 , Fig 16 it can be observed that the contribution of each generated feature coincides unmistakably in both golden score ,annotated with orange color and predicted score for grammar, annotated with blue color. Thus, concluding the accuracy of the implemented AES system when it comes to predicting grammar score using said generated feature and its closeness to the golden grammar score.

CONCLUSION

The developed Automated Essay Scoring System will be able to improve the AES system[4] by adding aspects of grammatical error detection in the system.

Inclusion of BERT Text Embedding has added a strong base to our Automated Essay Scoring System as the context of words will be considered when textual data will be converted to its numerical equivalent for the task of Natural Language processing. Lastly, Grammar Error detection has added more

granularity to the task of scoring the essays thus bridging the gap between the computer-generated solution and Real-life score assigning.

REFERENCES

[1] "The Hewlett Foundation: Automated Essay Scoring." Available at https://kaggle.com/competitions/asap-aes.

[2] K. Taghipour and H. T. Ng, "A Neural Approach to Automated Essay Scoring," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 1882–1891, Association for Computational Linguistics, 2016.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, 2019.

[4] P. Wangkriangkri, C. Viboonlarp, A. T. Rutherford, and *E*. Chuangsuwanich, "A Comparative Study of Pretrained Language Models for Automated Essay Scoring with Adversarial Inputs," in *2020 IEEE REGION 10 CONFERENCE (TENCON)*, (Osaka, Japan), pp. 875–880, IEEE, Nov. 2020.

[5] E. Mayfield and A. W. Black, "Should You Fine-Tune BERT for Automated Essay Scoring?," in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, (Seattle, WA, USA → Online), pp. 151–162, Association for Computational Linguistics, 2020.

[6] A. Warstadt, A. Singh, and S. R. Bowman, "Neural Network Acceptability Judgments," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 625–641, Nov. 2019.

[7] B. Shmueli, "Matthews Correlation Coefficient is The Best Classification Metric You've Never Heard Of," May 2020. Available at https://bit.ly/3AHIvNy.

[8] Manu Vishwakarma, "Introduction to BERT," 2019. Available at https://manu-vishwakarma.github.io/ manu-vishwakarma/BERT/.

[9] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," Feb. 2019. arXiv:1803.08375 [cs, stat].

[10] "TextBlob: Simplified Text Processing — TextBlob 0.16.0 documentation." Available at https://textblob.readthedocs.io/en/dev/.

[11] M. Näther, "An In-Depth Comparison of 14 Spelling Correction Tools on a Common Benchmark," in *Proceedings of the 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 1849–1857, European Language Resources Association, May 2020.

[12] "Splitting sentences into clauses." Available at https://subscription.packtpub.com/book/data/9781838987312/2/ch02lvl1sec13/splitting-sentences-into-clauses.

[13] "The Corpus of Linguistic Acceptability (CoLA)." Available at https://nyu-mll.github.io/CoLA/.

[14] K. Pykes, "Cohen's Kappa," Jan. 2021. Available at https://towardsdatascience.com/cohens-kappa-9786ceceab58.

[15] "Building a Grammar Correction Python App with Gramformer and Gradio." Available at https://bit.ly/3KfEUJG.

[16] A. A. Yusuf, N. A. Nwojo, and M. M. Boukar, "Basic dependency parsing in natural language inference," in *2017 13th International Conference on Electronics, Computer and Computation (ICECCO)*, (Abuja, Nigeria), pp. 1–4, IEEE, Nov. 2017.

[17] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, p. 6, Dec. 2020.

[18] H. Chimingyang, "An Automatic System for Essay Questions Scoring based on LSTM and Word Embedding," in *2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)*, (Shenyang, China), pp. 355–364, IEEE, Nov. 2020.

[19] R. Horev, "BERT Explained: State of the art language model for NLP," Nov. 2018. Available at https://bit.ly/2KCNELj.

[20] S. SHARMA, "Activation Functions in Neural Networks," July 2021. Available at https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.