A Multitask Learning System for Trait-based Automated Short Answer Scoring

Dadi Ramesh[1*], Suresh Kumar Sanampudi[2]
School of Computer Science and Artificial Intelligence, SR University, Warangal, India[1]
Research Scholar in JNTU, Hyderabad, India[1]
Department of Information Technology-JNTUH College of Engineering Jagitial,
Nachupally, (Kondagattu), Jagtial dist Telangana, India[2]

***Abstract*—Evaluating students' responses and providing feedback in the education system is widely acknowledged. However, while most research on Automated Essay Scoring (AES) has focused on generating a final score for given responses, only a few studies have attempted to generate feedback. These studies often rely on statistical features and fail to capture coherence and content-based features. To address this gap, we proposed a multitask learning system that can capture linguistic, coherence, and content-based features with Bidirectional Encoder Representations from Transformers (BERT) sentence by sentence and generate overall essay and trait scores. Our proposed system outperformed other existing models, achieving Quadratic Weighted Kappa (QWK) scores of 0.766, 0.69, and 0.701 compared to human rater scores. We evaluated our model on the Automated Student Assessment Prize (ASAP) Kaggle and operating system (OS) data set. When compared with other prescribed models proposed to multitask learning system is a promising step towards more effective and comprehensive writing assessment and feedback.**

***Keywords*—Sentence embedding; coherence; LSTM; short answer scoring; trait score**

I. INTRODUCTION

Evaluating student responses and providing feedback can improve the student's learning abilities in the education system. However, while AES has been a research focus in recent years, most studies have concentrated on generating a final score for student responses rather than providing feedback. In a few studies like [6, 13, 14, 15 and 16] systems, they did not attempt to generate feedback. However, these approaches extract statistical features for the final score and trait score generation, which did not fully capture students' responses' coherence and content-based features.

There are two main ways to provide feedback to students on their writing: gaze behavior by Mathias and Bhattacharyya [13, 14] and providing trait scores [6, 17, 14, 16, 25 and 26]. Gaze behavior refers to analyzing the visual behaviors of readers as they read a text, such as eye movement patterns. This methodology offers insights into readability, syntax, and fluency within the writing. However, trait scores evaluate specific writing attributes like organization, word choice, and coherence. These traits furnish more intricate insights into a student's writing than a simple overarching score.

Despite the strides made in deep learning and natural language processing, furnishing feedback on aspects such as organization, word choice, and syntax can benefit student learning more than just presenting a general score. As Woods [23] exemplified, this form of feedback equips students with a deeper comprehension of how to enhance their writing competencies holistically. For instance, feedback on organization assists students in grasping effective structuring techniques for their compositions. In contrast, feedback on word choice can help students select appropriate words to convey their ideas more clearly.

To accurately assess the student's response and provide comprehensive scoring and feedback, extracting both semantic and linguistic features from the text is essential. Relying solely on statistical features like Term Frequency and Inverse Document Frequency (TF-IDF), a bag of words, and N-gram models may not sufficiently capture the content and coherence of the student's answer. These traditional methods mainly focus on the frequency and distribution of individual words or phrases. In contrast, modern natural language processing (NLP) models such as word2vec [12] and Global vectors for word representation [7] can effectively capture semantic features by representing words in a continuous vector space. However, it is worth noting that these models like [1, 3, 4, and 13] primarily operate at the word level and may encounter difficulties when handling complex or polysynthetic words.

Deep learning models can be employed to address the need for capturing content, coherence, and maintaining the sequence of words. Recurrent Neural Networks (RNNs), particularly models like Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs), are commonly used for sequential data processing in NLP. These models like [13, 20, 21, 22 and 24], can analyze text at the sentence or paragraph level, considering the contextual information preceding words provide. Ridley et al. in [18] implemented a system for cross prompt essay scoring with semi supervised learning; furthermore, maintaining an internal state can capture dependencies and semantic relationships between words.

A deep learning model trained on a suitable dataset can be developed in the context of grading and providing feedback on student responses. This model would take the student response as input, process it using an RNN or similar architecture, and generate a final score and trait score. The training data for such a model ideally includes labeled examples of student responses paired with their corresponding scores and traits. This way, the model can learn to recognize patterns and

*Corresponding Author

associations between the input text and the desired output scores.

Overall, combining advanced NLP techniques, deep learning models, and appropriate training data can help extract content and coherence features while maintaining the sequence of words, thereby enabling the generation of accurate scores and feedback on student responses.

A. Contribution

- Our AES system captures sentence-level features from responses, enhancing essay analysis by highlighting key traits and patterns at this granular level. These features provide valuable insights into essay quality and structure.
- Employing LSTM, a type of recurrent neural network, we assign scores to essays. LSTM's strength in capturing context and relationships among sentences makes it ideal for modeling and analyzing essays.
- Our AES system produces three scores - overall, organization, and word choice. This trio offers a multi-faceted evaluation of essays, assessing different writing aspects like coherence, logical flow, and vocabulary sophistication. Additionally, we compare our model against established approaches to demonstrate its superiority.
- We test it on two datasets, a public dataset and a domain-specific dataset. Testing on different datasets helps evaluate the generalizability of our model across different domains or essay topics. It demonstrates the versatility and adaptability of our AES system.

Organization: The remainder of the paper is organized as follows: Section II illustrates related works on various evaluation systems and challenges. Section III discusses the proposed model and the data set used for our models. Section IV discusses the results and analysis of our model on various factors and test cases. Finally, Section V discusses the conclusion and future work.

II. Related Work

Automated Essay Scoring (AES) has primarily focused on generating a final score for student responses rather than providing detailed feedback. This emphasis on scoring is often driven by the need for standardized assessment, where the primary goal is to assign a numerical score that reflects the quality of the essay. So many researchers worked on the final score generation for the given response. Various systems, such as those developed by [4, 5, 9, and 10] as well as [11] have adopted distinct approaches. These approaches involve combining different elements, such as statistical features and word-level attributes, and training machine learning or neural network models. However, a noteworthy aspect is that these methodologies need to effectively encapsulate the entirety of the content within the essays into their respective vectors.

Generating feedback is more challenging because it requires understanding the student's response's content, structure, and coherence. While some AES systems [2, 8, 16 and 23] provide generic feedback based on predefined patterns or rules, the quality and specificity of this feedback may be limited.

However, there has been increasing interest in developing AES systems that go beyond scoring and provide more meaningful feedback to students. With advancements in natural language processing and deep learning, researchers are exploring approaches to extract fine-grained linguistic and semantic features from essays, which can be used to provide personalized feedback.

Ridley et al. in [17, 18] introduced a method that utilizes a trait-attention mechanism and a multi-task architecture to predict student essays' overall and individual trait scores. They conducted extensive experiments on the ASAP dataset, specifically prompt-2, to demonstrate the effectiveness of their approach for prompt-specific trait scoring and cross-prompt AES methods. To optimize model performance, researchers integrated syntactical elements by applying POS embedding. They employed LSTM models for generating comprehensive scores encompassing overall performance and specific traits. Additionally, Mathias and Bhattacharyya in [15] proposed a neural network model targeting word-level semantic features, enhancing the granularity of assessment. The common thread in these approaches is using Long Short-Term Memory (LSTM) models to predict multiple trait scores, enabling intricate essay evaluation. This technique captures nuanced semantic information at the word level, shedding light on the multifaceted traits manifested within the essays.

Hussein et al. in [6] focused on the ASAP dataset, employing LSTM to generate trait scores. Their model was tailored explicitly, capturing relevant patterns and features for accurate score prediction.

Woods et al. in [23] employed standard logistic regression to derive trait scores, providing a comprehensive assessment avenue. Although simpler than LSTM, logistic regression still yields valuable predictions and insights into trait scores.

Ohta et al. in [16] introduced a versatile model generating various scores, spanning overall evaluation, organizational development, and language proficiency. Their approach likely amalgamates techniques, potentially encompassing deep learning models such as LSTM, to predict and appraise different essay facets holistically.

These studies underscore diverse techniques and methodologies for forecasting trait scores in student essays. Each approach contributes unique insights, leveraging attention mechanisms, multi-task architectures, syntactical and statistical attributes, and the power of LSTM and logistic regression models for robust evaluation. In addition, these diverse methods contribute to the field by providing various options for assessing and providing feedback on different aspects of essay writing.

III. Methodology

We proposed that the AES system incorporates sentence-based text embeddings and LSTM to capture essay coherence and content. This model is implemented on top of [19] it generates multiple scores, including overall, organization, and word choice scores, providing a comprehensive evaluation.

Using both standard and domain-specific datasets strengthens the credibility and generalizability of our system. In addition, robustness testing ensures its resilience in handling adversarial responses. The architecture diagram Fig. 1 visually represents the system's components and integration, facilitating replication and implementation.

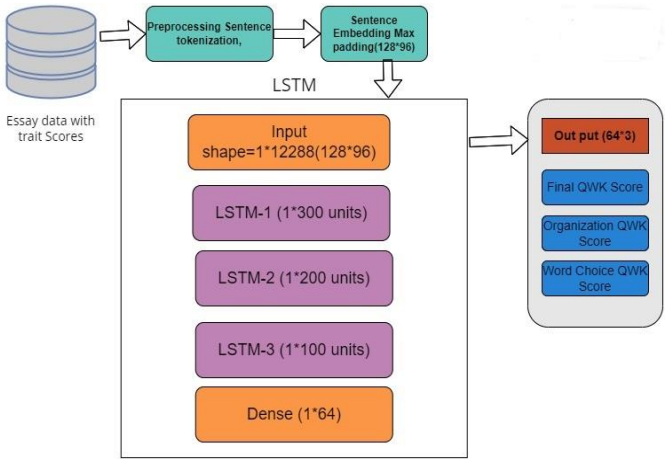

Fig. 1. Architecture of content based essay scoring system with LSTM.

A. Data Set

In the context of our AES systems, we used the ASAP Kaggle dataset. This dataset comprises 12,978 essays written by students in grades 8 to 10. The essays were generated in response to eight different prompts provided to the students. Every prompt comprises a collection of 1500 or more essays, all of which have been evaluated by two individual raters. Prompts 3, 4, 5, and 6 pertain to essays that rely on specific sources for their content, whereas the remaining prompts fall under the "others" category.

For our trait-based essay scoring approach, we specifically focused on Prompt 2 during our study. This allowed us to analyze and evaluate the essays based on specific traits and criteria relevant to that prompt. Table I and Table II exemplifies a detailed description of the essay dataset, including the number of essays, prompts, and raters involved.

TABLE I. AUTOMATED STUDENT ASSESSMENT PRIZE (ASAP) DATA SET FOR ESSAY SCORING

Prompt _id	Prompt wise total number of essays	Average Number of words in an essay	Score range (min-max)
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

TABLE II. OPERATING SYSTEM DATA SET (HTTPS://GITHUB.COM/RAMESHDADI/OS-DATA_1-SET-FOR-AES)

Prompt-id	Prompt	Prompt wise number of essays	Prompt wise maximum number of sentences	Rating range (min to max)
1	Explain about operating system?	516	23	0-5
2	Explain the advantages of a multiprocessor system?	596	21	
3	Explain how operating system handles multiple tasks at a time?	312	19	
4	Difference between single processor and multiprocessor operating systems?	513	19	
5	Explain different scheduling algorithm?	453	15	

In order to evaluate the performance of AES systems on domain-specific essays in the field of operating systems (OS), we created a custom dataset in addition to the ASAP dataset. This dataset was purposely crafted to address the field of operating systems. We formulated five fundamental inquiries concerning operating systems and then distributed these inquiries as assignments to students across various engineering colleges.

Upon gathering the responses, we meticulously eliminated duplicated or repeated submissions, culminating in a dataset comprising 2390 unique responses from 626 students. To ensure the dataset's dependability and excellence, we engaged in the expertise of two subject matter specialists. These professionals evaluated each essay, assigning scores for overall impression, organization, and word choice. This approach provided thorough and detailed evaluations of the students' written reactions.

By incorporating this tailor-made dataset, we intend to streamline the assessment of AES systems in their proficiency and accuracy when evaluating essays that specifically revolve around the operating systems realm. Furthermore, this dataset is a valuable asset for honing and optimizing AES systems within this domain.

The agreement between the two human raters was assessed using the QWK score. The computed QWK scores for the overall impression, organization, and word choice were 0.842, 0.879, and 0.912, respectively. These scores indicate a substantial agreement between the raters.

Table II illustrates a detailed description of the OS dataset, presenting information about the number of responses, students, and expert evaluators involved. Fig. 2 illustrates the agreement between the two raters, further illustrating the consistency in their evaluations.

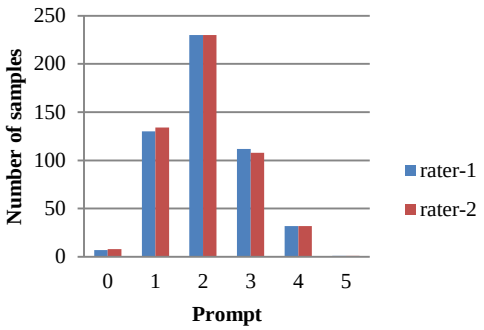

Fig. 2. Agreement between rater-1 and rater -2 (organization score for OS dataset).

B. Sentence Embedding

Converting text into vectors that effectively capture context and semantics is a complex task in natural language processing (NLP). Conventional embedding methods such as word2vec and GloVe primarily concentrate on converting text into word vectors. However, these methods come with certain constraints. They must thoroughly account for the words around a particular word and how their meanings change in different contexts. Furthermore, these techniques encounter difficulties when dealing with words with multiple meanings (polysemous words), which can result in a lack of accuracy in capturing the intended sense of the word.

Furthermore, approaches such as averaging word vectors to derive sentence vectors cannot capture the nuanced information in the original sentence. This oversimplification needs to be revised to include the intricate relationships and interactions between words, leading to a loss of important contextual details.

To address these challenges, more advanced techniques in NLP are being developed. These techniques aim to overcome the limitations of traditional methods by capturing a richer representation of text that incorporates both context and semantics more comprehensively. Our model utilizes Sentence BERT, a sentence embedding technique, to address these limitations. Sentence BERT introduces a dynamic approach to converting essays into vectors, considering the contextual and semantic aspects of individual sentences. Unlike traditional embedding techniques, Sentence BERT's vector representation captures a more comprehensive understanding of the original text.

The process begins by removing special symbols like "@" and "#" from the essays and tokenizing them into sentences. The ASAP and OS datasets have specific limitations on the maximum number of sentences per essay. The maximum number of sentences per essay for the ASAP dataset is 96, while for the OS dataset, it is 23.

To obtain sentence embeddings using a pre-trained transformer model, such as Sentence BERT, each sentence is transformed into a 128-dimensional vector representation. As a result, for an essay from the ASAP dataset, there will be 96 * 128-dimensional vectors, considering the maximum number of sentences. Similarly, for an essay from the OS dataset, there will be 23 * 128-dimensional vectors based on the respective maximum number of sentences.

Finally, to ensure consistent dimensions, all essays are padded to have 96 * 128-dimensional vectors for the ASAP dataset and 23 * 128-dimensional vectors for the OS dataset. The maximum number of sentences in each dataset determines the padding size.

C. Model

To capture the coherence, cohesion, and linguistic features of the essay, we embedded all the sentences of the essays without removing stop words. So it allows the model to consider the entire text and capture the overall coherence of the essay.

In order to handle the sequence of sentence embeddings, we utilized LSTM (Long Short-Term Memory), a type of recurrent neural network (RNN). LSTM is designed to effectively process sequential information while retaining the memory of previous inputs.

The LSTM architecture incorporates various gates, as depicted in Fig. 3. These gates, including the input gate equation [2], forget gate [1], output gate [3], and context gate [4], collaborate to facilitate the processing and storage of long-term dependencies necessary for feature extraction.

The input gate controls the flow of information into the memory cell, while the forget gate determines which information to discard from the previous cell state. The output gate regulates the output information from the memory cell, and the context gate manages the update of the memory cell content.

$$f_t = \sigma g(w_f x_t + u_f h_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma g(w_i x_t + u_i h_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma g(w_o x_t + u_o h_{t-1} + b_o) \quad (3)$$

$$C_t = tanh(w_c x_t + u_c h_t + b_c) \quad (4)$$

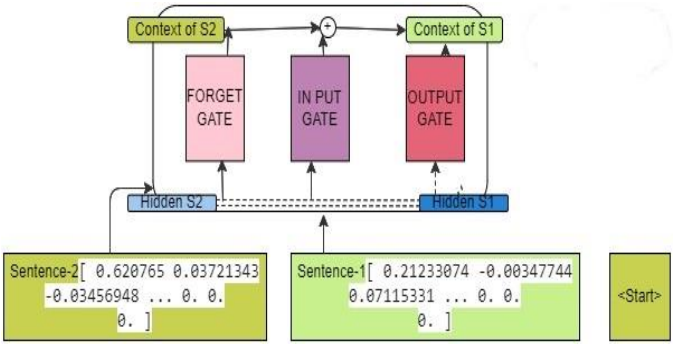

Fig. 3. Contexts generation from sentences in LSTM.

D. Implementation and Training LSTM

In our methodology, we started by turning essays into organized sets of numbers using a technique called Sentence BERT. These sets were then made consistent by adding extra information and aligning them with the largest essay size of 96 by 128. After this, we transformed these sets of numbers into

another type of organized sets with three dimensions, getting them ready for neural network training.

To construct our LSTM model, we devised a configuration comprising five tiers of LSTM units intricately assembled. Each tier encompassed components that facilitated the absorption of information, dissemination of information, and context management. This structural design proved pivotal in monitoring the interconnections spanning distinct sections of the essays.

To enhance the efficacy of our model, we adopted the RMSprop optimization technique, concentrating on minimizing the mean discrepancy between our forecasts and the actual results. To counteract the risk of our model fixating excessively on idiosyncrasies within the training data, we integrated a mechanism that intermittently deactivated specific model segments during the training phase. Furthermore, we established a predetermined pace for how our model assimilates knowledge from the data. Our model's response to data inputs was governed by a selected mathematical function termed ReLU. Throughout the model training process, we implemented a strategy known as 5-fold cross-validation. This divided our sets of essay information into separate groups for training, testing, and validating, with a specific ratio assigned to each for both the ASAP and OS datasets.

We trained our model for different amount of time (10, 15, 20, and 35 times), trying to find the best settings, and then we checked how well it performed. We used a QWK measure to see how close the model's ratings were to human ratings, an essential standard for automated essay scoring.

For each round of cross-validation, we calculated the QWK score. Finally, we chose the model that worked the best on the training data to make predictions on the test data. A graph (see Fig. 5) showed how the model learned and did on new data where batch size 24, demonstrating that it learned well without getting too stuck on the training details.

We are applied the same hyperparameters and cross-validation technique to ensure consistency in our experimentation and evaluation process between the ASAP and OS datasets. This allows for a fair comparison and assessment of our model's performance on different essay datasets.

IV. Result and Discussions

We conducted experiments using the ASAP and OS datasets to develop trait-based AES (Automated Essay Scoring) systems. Our proposed model exhibited the best results on both datasets. Furthermore, we approached the training of our model on a prompt-by-prompt basis and subsequently computed the corresponding training and validation losses. This process is visualized in Fig. 4 and Fig. 5. The figure illustrates the learning process for batch size 32, which is getting overfitted after some iteration. But from Fig. 5, when we used a batch size of 16, the training and validation losses consistently decreased and overlapped each other.

We achieved superior results when comparing our proposed models to other baseline models, specifically on prompt-2 of the ASAP dataset. We used the average QWK (Quadratic Weighted Kappa) score for each trait. As shown in Table III, our sentence embedding-LSTM model outperformed other models and demonstrated consistency with human rater scores. However, the integrated and word embedding models could have effectively captured sentence coherence. It is worth noting that while neural networks tend to achieve high QWK scores, they may not fully capture the text's coherence. The models implemented with word embeddings required maintaining word order, which could impact coherence. Additionally, word embedding models may struggle with polysemous words, potentially leading to a lack of coherence in the text.

Our proposed model performed well on the Domain-specific data set, specifically regarding the final, organization, and word choice scores. This indicates that our model is robust and consistently delivers good results.

Table IV compares our proposed model and other approaches on the Domain-specific data set, demonstrating the superiority of our model across multiple scoring criteria. In addition, the consistent performance of our model suggests its reliability and effectiveness in evaluating essays within the specific domain. Fig. 6 illustrates the comparison of the actual score and predicted scores of our models. From this, we can observe that both colors are overlapped maximum, which indicates that our model predicted the correct scores for test data.

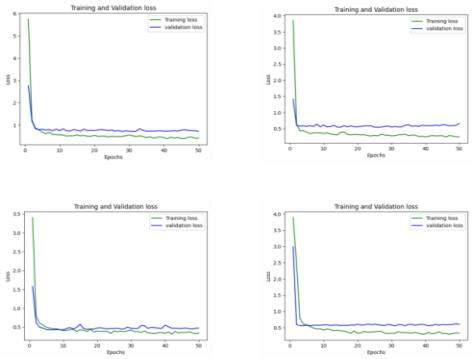

Fig. 4. Prompt wise training and validation loss of sent-LSTM model (batch size=32).

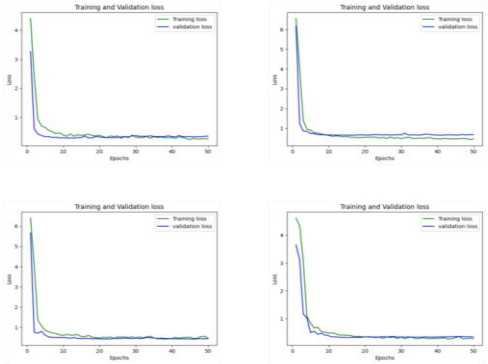

Fig. 5. Prompt wise training and validation loss of sent-LSTM model (batch size=16).

TABLE III. NEURAL NETWORK MODEL HYPER PARAMETERS AND TRAINED VALUES

Layer	parameter	Value
Embedding	Sentence Embedding (BERT)	128 size vectors for each sentence
Input and output	Input size output	(1,96,128), (1, 23,128) 1*3
LSTM layers	No of layers	5
LSTM Units	LSTM Units	300
Hidden	Hidden units	200,100
Drop out	Dropout rate Recurrent Drop out	0.4 0.5
Others	Epochs Batch size Learning Rate Optimizer	35 32 0.001 Adam

TABLE IV. COMPARISONS OF RESULTS ON ASAP DATA SET PROMPT-2(QWK SCORE) WITH PRESCRIBED MODELS

System	Overall score	Organizatio	Word choice	ASAP dataset	Remarks
[17]	0.453	0.243	0.416	All prompts	Word embedding model
[14]	0.563	0.551	0.531	Prompt-2	
[6]	0.402	0.256	0.402	All prompts	
[1]	0.600	0.570	0.583	Prompt-2	
[4]	0.617	0.623	0.630	Prompt-2	
Sent-LSTM model	0.691	0.677	0.679	Prompt-2	Sentence embedding model
Sent-LSTM model	0.672	0.66	0.673	OS-data set	Sentence embedding model

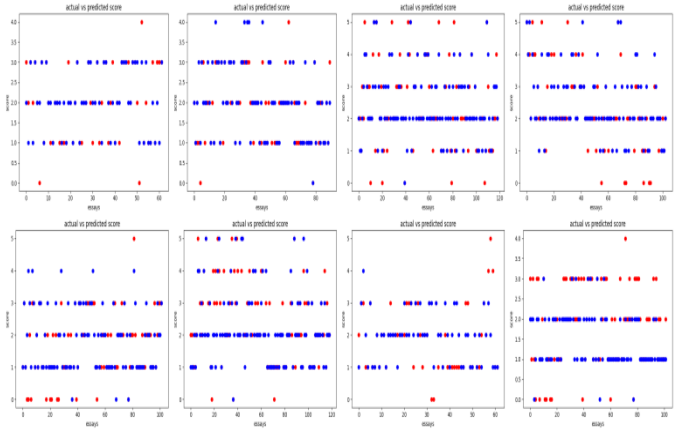

Fig. 6. Comparisons of actual and predicted score of sent-LSTM, top row batch size are 16, and bottom row batch size is 32.

V. CONCLUSION

We introduced a novel approach for an AES system focusing on trait-based assessment. To capture the coherence patterns between sentences in an essay, the model employed a preprocessing step where each sentence was embedded individually. This sequence-to-sequence approach allows the model to capture the overall coherence and flow of ideas within the essay. The embedded sentences were then fed into a recurrent neural network, specifically an LSTM, for training.

In our study, we compared our Sentence Embedding-LSTM model with baseline models commonly used in AES. The results showed that our proposed model performed well and outperformed the other baseline models in terms of its ability to capture coherence. So, our approach successfully addressed the challenge of maintaining coherence in the generated essays.

Leveraging sentence embedding and training on recurrent neural networks model demonstrated its effectiveness in capturing the overall organization and coherence of the essays. However, our approach has the potential to provide more reliable and accurate automated scoring while preserving the coherence of the generated texts.

In the future, we will implement a system to provide all traits like grammar and sentence organization and provide the students Qualitative feedback in text format. And we also improve model performance.

ACKNOWLEDGMENT

We thank SR University and JNTU college of Engineering Jagitial, students, and faculty for collecting and assessing OS dataset.

REFERENCES

[1] Cozma, M., Butnaru, A. M., & Ionescu, R. T. (2018). Automated essay scoring with string kernels and word embeddings. arXiv preprint arXiv:1804.07954. https://doi.org/10.48550/arXiv.1804.07954

[2] Carlile, W., Gurrapadi, N., Ke, Z., & Ng, V. (2018, July). Give me more feedback: Annotating argument persuasiveness and related attributes in student essays. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 621-631). DOI:10.18653/v1/P18-1058

[3] Dasgupta, T., Naskar, A., Dey, L., &Saha, R. (2018, July). Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications (pp. 93-102). DOI:10.18653/v1/W18-3713

[4] Dong F, Zhang Y, &Yang J (2017) Attention-based recurrent convolutional neural network for automatic essay scoring. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017) p 153–162. DOI: 10.18653/v1/K17-1017

[5] Fernandez, N., Ghosh, A., Liu, N., Wang, Z., Choffin, B., Baraniuk, R., & Lan, A. (2022). Automated Scoring for Reading Comprehension via In-context BERT Tuning. arXiv preprint arXiv:2205.09864. https://doi.org/10.48550/arXiv.2205.09864

[6] Hussein, M. A., Hassan, H. A., & Nassef, M. (2020). A trait-based deep learning automated essay scoring system with adaptive feedback. International Journal of Advanced Computer Science and Applications, 11(5). DOI : 10.14569/IJACSA.2020.0110538

[7] Pennington, J., Socher, R., & Manning, C.D. (2014). GloVe: Global Vectors for Word Representation. *Conference on Empirical Methods in Natural Language Processing*. pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.DOI:10.3115/v1/D14-1162

[8] Ke, Z., Carlile, W., Gurrapadi, N., & Ng, V. (2018, July). Learning to Give Feedback: Modeling Attributes Affecting Argument Persuasiveness in Student Essays. In IJCAI (pp. 4130-4136). https://doi.org/10.24963/ijcai.2018/574

[9] Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019, July). Get it scored using autosas—an automated system for scoring short answers. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 9662-9669). DOI: https://doi.org/10.1609/aaai.v33i01.33019662

[10] Kabra, A., Bhatia, M., Kumar, Y., Li, J. J., Jin, D., & Shah, R. R. (2020). Calling Out Bluff: Evaluation Toolkit for Robustness Testing Of Automatic Essay Scoring Systems. *arXiv preprint arXiv:2007.06796*.

[11] Lun, J., Zhu, J., Tang, Y., & Yang, M. (2020). Multiple Data Augmentation Strategies for Improving Performance on Automatic Short Answer Scoring. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(09), 13389-13396. https://doi.org/10.1609/aaai.v34i09.7062

[12] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

[13] Mathias S, Bhattacharyya P (2018) ASAP++: Enriching the ASAP automated essay grading dataset with essay attribute scores. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)

[14] Mathias, S., & Bhattacharyya, P. (2020, July). Can neural networks automatically score essay traits?. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications (pp. 85-91). DOI:10.18653/v1/2020.bea-1.8

[15] Mathias, S., Murthy, R., Kanojia, D., Mishra, A., & Bhattacharyya, P. (2020). Happy are those who grade without seeing: A multi-task learning approach to grade essays using gaze behaviour. arXiv preprint arXiv:2005.12078.

[16] Ohta, R., Plakans, L. M., & Gebril, A. (2018). Integrated writing scores based on holistic and multi-trait scales: A generalizability analysis. Assessing Writing, 38, 21-36. doi.org/10.1016/j.asw.2018.08.001.

[17] Ridley, R., He, L., Dai, X., Huang, S., & Chen, J. (2020). Prompt Agnostic Essay Scorer: A Domain Generalization Approach to Cross-prompt Automated Essay Scoring. arXiv preprint arXiv:2008.01441. https://doi.org/10.48550/arXiv.2008.01441

[18] Ridley, R., He, L., Dai, X. Y., Huang, S., & Chen, J. (2021, May). Automated cross-prompt scoring of essay traits. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 15, pp. 13745-13753). https://doi.org/10.1609/aaai.v35i15.17620

[19] Ramesh, D., & Sanampudi, S. K. (2022, November). Coherence Based Automatic Essay Scoring Using Sentence Embedding and Recurrent Neural Networks. In Speech and Computer: 24th International Conference, SPECOM 2022, Gurugram, India, November 14–16, 2022, Proceedings (pp. 139-154). Cham: Springer International publishing. https://doi.org/10.1007/978-3-031-20980-2_13

[20] Singh, S., Pupneja, A., Mital, S., Shah, C., Bawkar, M., Gupta, L. P., ... & Shah, R. R. (2023). H-AES: Towards Automated Essay Scoring for Hindi. arXiv preprint arXiv:2302.14635.

[21] Taghipour, K., & Ng, H. T. (2016, November). A neural approach to automated essay scoring. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1882-1891). DOI:10.18653/v1/D16-1193.

[22] Uto, M., & Okano, M. (2021). Learning automated essay scoring models using item-response-theory-based scores to decrease effects of rater biases. IEEE Transactions on Learning Technologies, 14(6), 763-776.

[23] Woods, B., Adamson, D., Miel, S., & Mayfield, E. (2017, August). Formative essay feedback using predictive scoring models. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 2071-2080). https://doi.org/10.1145/3097983.3098160

[24] Wang, Y., Wang, C., Li, R., & Lin, H. (2022). On the Use of BERT for Automated Essay Scoring: Joint Learning of Multi-Scale Essay Representation. arXiv preprint arXiv:2205.03835. https://doi.org/10.48550/arXiv.2205.03835.

[25] Wang, J., Chen, J., Ou, X., Han, Q., & Tang, Z. (2023). Multi-level Feature Fusion for Automated Essay Scoring.

[26] Yang, R., Cao, J., Wen, Z., Wu, Y., & He, X. (2020, November). Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 1560-1569). DOI:10.18653/v1/2020.findings-emnlp.141.