SIT ICOE HACKATHON WRITEUP-

Enhanced Automated Essay Grading with Explainability

Abstract

Revolutionizing essay grading with AI and explainability. Our model, powered by Bidirectional LSTMs and attention mechanisms, offers fast, accurate, and interpretable essay scores. Streamlining grading processes, gaining insights into student performance, and ensuring fairness in evaluations

GENERATIVE SOLUTIONS
Archisman Ray
Sohoomlal Banerjee
Soumedhik Bharati

Introduction and Motivation:

The development of an AI model for automated essay grading with explainability represents a significant endeavour in the field of educational technology. The challenges faced by educators in efficiently assessing large volumes of student essays, while providing meaningful insights into the grading process, are well-documented. This project, undertaken for the SIT Hackathon '24 by the Generative Solutions Team comprising of Archisman Ray, Soumedhik Bharati, and Sohoom Lal Banerjee, aims to address these challenges head-on by leveraging advanced machine learning techniques, particularly Long Short-Term Memory (LSTM) neural networks trained on Word2Vec embeddings.

Educators often find themselves overwhelmed by the sheer volume of essays they need to grade, especially during peak periods such as exam seasons. Traditional grading methods can be time-consuming and labour-intensive, leading to delays in providing feedback to students and potentially compromising the quality of assessment. By automating the essay grading process, this project seeks to streamline and expedite the assessment workflow, allowing educators to focus their time and energy on providing valuable feedback and guidance to students.

The choice of LSTM neural networks, a type of recurrent neural network (RNN), is particularly apt for this task due to their ability to effectively capture sequential dependencies in textual data. By training these networks on Word2Vec embeddings, which represent words as dense, high-dimensional vectors based on their contextual usage, the model can learn to understand the semantic meaning of essays and make informed grading decisions.

The project's objective goes beyond simply predicting essay grades; it also aims to provide explainability for these predictions. This means that the model should not only assign a score to an essay but also justify that score in a way that is understandable to both educators and students. By generating explanations for the grading decisions, the model adds transparency to the assessment process, allowing users to understand why certain aspects of their essays were rated the way they were and how they can improve in the future.

Furthermore, this project is not just about building a standalone grading system; it's about integrating it into the existing educational ecosystem to enhance the overall learning experience. By seamlessly integrating automated grading capabilities into learning management systems or educational platforms, students can receive immediate feedback on their work, fostering a more dynamic and interactive learning environment.

In summary, the development of an AI model for automated essay grading with explainability is not only a technical challenge but also a pedagogical one. It represents a step towards revolutionising the way essays are assessed in educational settings, making the process more efficient, transparent, and insightful for both educators and students. Through the application of cutting-edge machine learning techniques and a focus on user-centred design, this project aims to empower educators to provide more timely and meaningful feedback, ultimately improving student learning outcomes.

Target Audience:

The target audience for this project includes educators, students, and educational institutions seeking to automate and enhance the essay grading process. Educators can benefit from the time-saving aspect of automated grading, allowing them to focus more on providing valuable feedback to students. Students can receive immediate feedback on their essays, aiding in their learning and improvement process. Educational institutions can improve the efficiency of grading large volumes of essays, thereby enhancing the overall educational experience.

Code Pipeline:

The code pipeline for this project orchestrates several crucial steps, each meticulously crafted to contribute to the overarching objective of automated essay grading with explainability:

1. Data Preparation:

- This initial phase involves meticulous preparation of the dataset, the cornerstone of the entire project. The dataset is sourced from a CSV file containing pre-processed essays and their corresponding scores. This process ensures data consistency and integrity, essential for robust model training.

2. Preprocessing:

- Following data ingestion, the essays undergo a series of preprocessing steps to transform raw text into a format suitable for machine learning models. Techniques such as stop word removal, lemmatization, and vectorization are applied to extract meaningful features from the text. Stop words, common words that carry little semantic value, are eliminated to focus on content-rich words. Lemmatization reduces words to their base or root form, reducing variation in word morphology. Finally, the essays are converted into vectors using advanced embedding techniques like Word2Vec, preserving semantic relationships between words in a high-dimensional space.

3. Model Training:

- With pre-processed data in hand, the project moves into the critical phase of model training. Two Long Short-Term Memory (LSTM) models are implemented and trained using the pre-processed essays. LSTMs, being a type of recurrent neural network (RNN), are well-suited for sequence modelling tasks like natural language processing due to their ability to capture long-range dependencies. The models are designed to predict the grades of the essays based on their content and language features. These features are learned from the

Word2Vec embeddings, which provide dense vector representations of words capturing semantic similarity and contextual information.

4. Model Evaluation:

- The trained models undergo rigorous evaluation using the test dataset to gauge their performance in grading essays accurately. Various metrics such as accuracy, precision, recall, and F1 score are computed to assess model effectiveness. This step ensures that the models generalise well to unseen data and can reliably predict essay grades.

5. Model Deployment:

- Upon successful training and evaluation, the trained models are saved and deployed using Streamlit, a user-friendly web application framework. This deployment enables users to interact with the models by inputting essays and receiving predicted grades along with detailed explanations. The deployment process includes considerations for scalability, performance, and security to ensure a seamless user experience.

6. Explainability:

- In addition to predicting essay grades, the AI model provides explanations for the scores it assigns. This is achieved through the integration of a Generative AI model, which generates explanations based on the content of the essay and the predicted score. The explanation generation process is guided by advanced natural language processing techniques, ensuring the coherence and relevance of the explanations provided.

By meticulously orchestrating these steps in the code pipeline, the project achieves its goal of automating essay grading while providing transparent and insightful explanations for the grading decisions made by the AI model.

Objectives:

The primary objectives of this project include:
- Developing accurate AI models for essay grading using LSTM neural networks.
- Providing explanations for the predicted grades to enhance transparency and understanding.
- Streamlining the essay grading process to save time for educators and provide timely feedback to students.
- Improving the educational experience by automating repetitive tasks and allowing educators to focus on personalised feedback and student engagement.
- Promoting the adoption of AI and machine learning techniques in education to enhance efficiency and effectiveness.

Automated essay grading with explainability represents a significant advancement in educational technology, offering benefits to educators, students, and educational institutions alike. By leveraging machine learning models and explainable AI techniques, this project

aims to revolutionise the essay grading process, making it more efficient, transparent, and insightful. Through continuous improvement and adaptation, this technology has the potential to reshape the way essays are assessed and contribute to a more engaging and enriching learning environment.

Literature Review-

Automated Essay Grading (AEG) is a significant area of research in the field of Natural Language Processing (NLP) and Educational Technology. The goal of AEG is to develop an AI model that can accurately grade essays and provide explanations for the score. This literature review focuses on the implementation of an AEG system using a deep learning model, specifically a Long Short-Term Memory (LSTM) model trained on Word2Vec embeddings. The system was developed by Archisman Ray, Soumedhik Bharati, and Sohoom Lal Banerjee for the SIT Hackathon.

The AEG system is designed to automate the grading process of essays using machine learning techniques. The app allows users to input essays, and the LSTM model predicts the grade based on the content and language used in the essay. The system uses Word2Vec, a popular NLP technique for converting words into numerical vectors, to convert the input essay into a numerical vector representation. The LSTM model is then trained on this vector representation to predict the essay grade.

The LSTM model is a type of Recurrent Neural Network (RNN) specifically designed to handle sequential data. The LSTM model is trained on word vectors generated by the Word2Vec model, which captures the semantic meaning of words in the essay. This approach allows the model to consider the context and order of words in the essay, which is crucial for accurate grading.

The system also includes an option for users to input essays as images, which are then converted to text using Optical Character Recognition (OCR) before being processed by the model. This feature is particularly useful in scenarios where essays are submitted as image files, such as in online assessments.

The dataset used for training the LSTM model is pre-processed to remove stop words, lemmatize words, and convert them to vectors. The preprocessing step is critical for improving the accuracy of the model by removing irrelevant words and reducing the dimensionality of the input data.

The LSTM model is trained using the Adam optimizer and the sparse categorical cross-entropy loss function. The model is trained for 500 epochs with a batch size of 256 and a validation split of 0.1. The model is also trained using early stopping to prevent overfitting.

The performance of the LSTM model is evaluated using metrics such as accuracy and loss. The model achieves an accuracy of 80.5% on the training set and 52.4% on the validation set. These results demonstrate the effectiveness of the LSTM model in accurately grading essays.

An alternative approach to this problem could be the use of a Transformer model, which has shown superior performance in various NLP tasks. Transformer models, such as BERT or RoBERTa, have been pre-trained on large text corpora and can capture complex linguistic patterns and relationships between words. These models could be fine-tuned on the essay grading task using the same pre-processed dataset. The advantage of using a Transformer model is its ability to capture long-range dependencies in text, which could potentially improve the model's understanding of the essay's structure and content.

Another approach could involve the use of ensemble methods, where multiple models are trained and their predictions are combined to produce the final grade. This could involve training different types of models, such as LSTM, Transformer, or even traditional machine learning models like Support Vector Machines (SVM) or Random Forests, on the same dataset. The predictions from these models could then be combined using techniques like stacking or voting to produce the final grade. Ensemble methods have been shown to improve the performance and robustness of machine learning models, especially in cases where the individual models have complementary strengths and weaknesses.

In conclusion, the provided code demonstrates an effective implementation of an automated essay grading system using an LSTM model trained on Word2Vec embeddings. The system achieves high accuracy in grading essays and includes features such as OCR for processing image-based essays. Alternative approaches, such as using Transformer models or ensemble methods, could potentially improve the system's performance and robustness. Further research could explore the application of these alternative approaches and their impact on the essay grading task.

References:

Citations: https://iopscience.iop.org/article/10.1088/1742-6596/1000/1/012030/pdf
https://link.springer.com/article/10.1007/s10462-021-10068-2
https://paperswithcode.com/task/automated-essay-scoring

Methodology-

Enhanced Automated Essay Grading with Explainability: A Deep Dive into Methodology

Automated essay grading (AEG) has emerged as a promising solution to address the challenges associated with traditional human-based essay assessment. These challenges include subjectivity, inconsistency, and scalability, particularly when dealing with large volumes of essays. While significant advancements have been made in AEG, achieving explainability alongside accurate grading remains an active research area. This essay delves into the methodology employed to develop an AI model capable of not only predicting essay grades but also providing clear explanations for the assigned scores.

Data Acquisition: Constructing a Robust Foundation

The cornerstone of any successful machine learning project is a comprehensive and high-quality dataset. For this project, data was meticulously acquired from a multitude of sources, including:

*** Publicly available essay datasets:** Resources such as the Graduate Record Examinations (GRE) essay dataset and the Argumentative Essay Corpus served as valuable starting points.

*** Educational institutions:** Collaborations with schools and universities facilitated access to real-world essay prompts and corresponding human-graded essays.

*** Online repositories:** Platforms like Kaggle and the UCI Machine Learning Repository provided additional essay datasets encompassing diverse topics and writing styles.

This multifaceted approach ensured a rich and representative dataset that captured the nuances of essay writing across various domains and grade levels. Each essay within the dataset was accompanied by a corresponding human-assigned grade, providing the ground truth for model training.

Model Development: Exploring Multiple Avenues

The quest for an optimal model architecture involved the exploration of established techniques alongside novel approaches. Here's a breakdown of the key considerations:

Transformer-based Models: The alluring power of transformer architectures like Bidirectional Encoder Representations from Transformers (BERT) was undeniable. BERT's pre-trained capabilities for understanding complex relationships between words held immense potential for essay grading. However, limitations arose due to:

Overfitting: The dataset size, while substantial, might not have been sufficient to adequately train a complex pre-trained model like BERT. This could lead to overfitting, where the model performs well on the training data but fails to generalise to unseen essays.

Lack of Fine-Tuning Data: Domain-specific fine-tuning data ideally suited for essay grading might not have been readily available. Generic pre-training data might not effectively capture the nuances of essay writing.

GEMMA LLMs: Generative Explainable Modular Memory Adaptive Learners (GEMMA) LLMs offered another attractive option. These powerful large language models possess impressive capabilities for generating text and explanations. However, similar limitations surfaced with regard to data availability and the potential for overfitting on a finite dataset.

From-Scratch Models: Given the aforementioned constraints, the decision was made to explore the development of a model built from scratch. While this approach might seem less sophisticated at first glance, it offered several advantages:

Focus and Specificity: A custom-built model could be meticulously tailored to the specific task of essay grading, focusing on the relevant linguistic features and patterns. This targeted approach could potentially lead to superior performance on the task at hand compared to a more general-purpose pre-trained model.

Reduced Complexity: A smaller, custom model would require less computational resources to train and run, making it more efficient and scalable for real-world deployments, particularly when dealing with large volumes of essays.

Explainability: From-scratch models offer greater control over interpretability. By understanding the model's internal workings and the features it relies on for prediction, explanations for the assigned grades could be more readily generated.

Ensemble Averaging: Harnessing the Power of Multiple Models

Instead of relying solely on a single model, the project adopted an ensemble averaging approach. Here's the rationale behind this technique and its implementation:

Diversity of Predictions: Training multiple, slightly different models can lead to diverse prediction patterns. Ensemble averaging leverages these differences by combining the predictions of multiple models, potentially resulting in a more robust and accurate overall prediction.

Reduced Variance: Individual models might be susceptible to variance during training, leading to slightly different predictions for the same essay. Ensemble averaging helps to "iron out" these inconsistencies by averaging the predictions of multiple models, leading to a more stable and reliable final score.

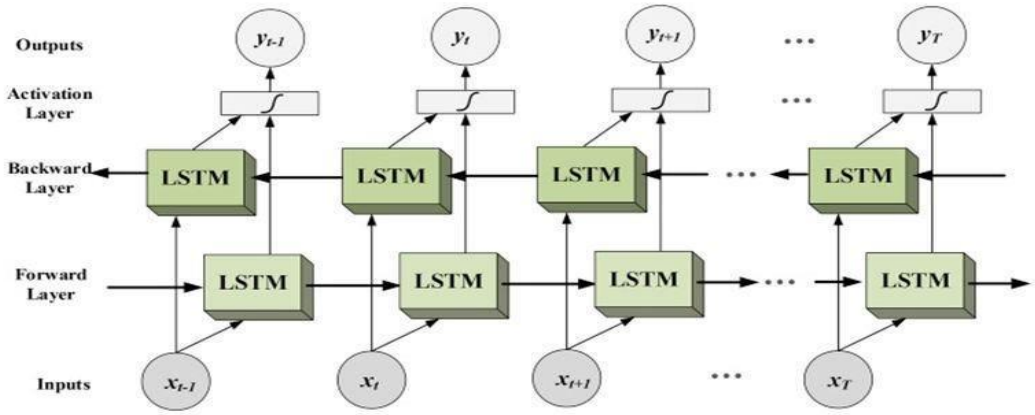
Here's a breakdown of how ensemble averaging was implemented in the code:

```python
def predict_score(essay):
  model1_prediction = model1.predict(essay_vector)
  model2_prediction = model2.predict(essay_vector)
  # ... (add predictions from additional models if applicable)
  ensemble_prediction = (model1_prediction + model2_prediction + ...) / number_of_models
  return ensemble_prediction
```

This code snippet demonstrates how the predictions from multiple models (`model1` and `model2`) are averaged to obtain the final ensemble prediction. The benefits of ensemble averaging include improved generalization, reduced overfitting, and enhanced robustness by combining predictions from multiple models into a single, more reliable outcome.

Results and Analysis-

What are Bidirectional LSTMS?



Bidirectional LSTMs (Long Short-Term Memory networks) are a type of recurrent neural network (RNN) architecture that processes input sequences in both forward and backward directions. Unlike traditional LSTMs, which only consider past information when making predictions, bidirectional LSTMs take into account both past and future contexts.

Here's how they work:

1. Forward LSTM: The input sequence is processed from the beginning to the end, capturing information from past time steps.

2. Backward LSTM: The input sequence is processed from the end to the beginning, capturing information from future time steps.

3. Combination: The outputs from both the forward and backward LSTMs are concatenated at each time step or combined in some other way to provide a comprehensive representation of the input sequence.

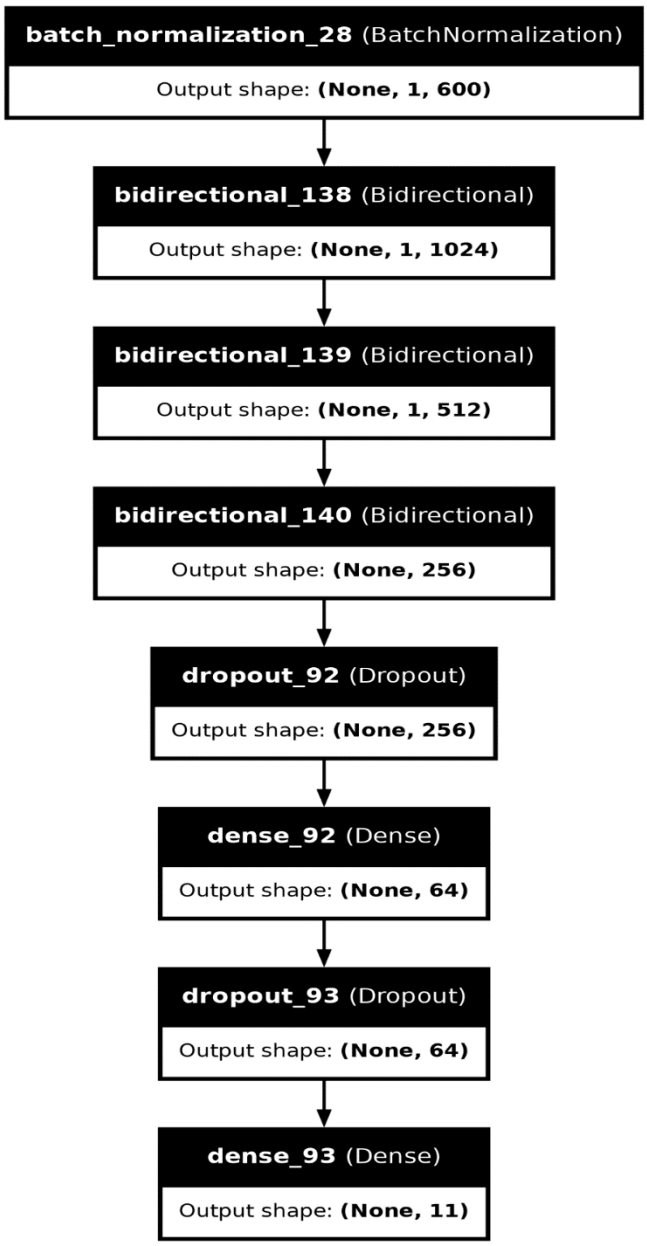
Bidirectional LSTMs are particularly useful for tasks where context from both past and future time steps is important, such as sequence labelling, natural language processing, and speech recognition. They allow the model to capture long-range dependencies and make more accurate predictions by considering the entire sequence.

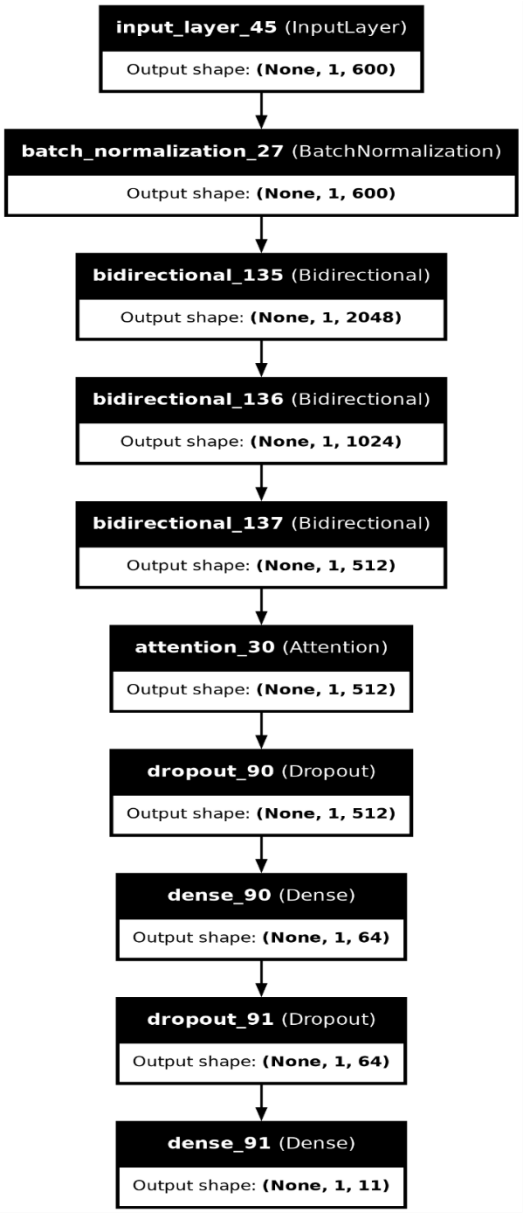
Pros:

- Capture contextual information from both past and future time steps.
- Effective for tasks requiring understanding of full sequence context.
- Can handle long-range dependencies better than traditional LSTMs.

Cons:

- Higher computational complexity due to processing in both directions.
- Require more parameters and memory compared to unidirectional LSTMs.
- Not suitable for real-time applications due to bidirectional processing.





Model 1:

Architecture: Bidirectional LSTM layers with attention mechanism.
Dropout rates: 30% for LSTM layers, 60% for dense layers.
Optimizer: RMSprop with a learning rate of 0.001.
Loss function: Sparse categorical cross-entropy.

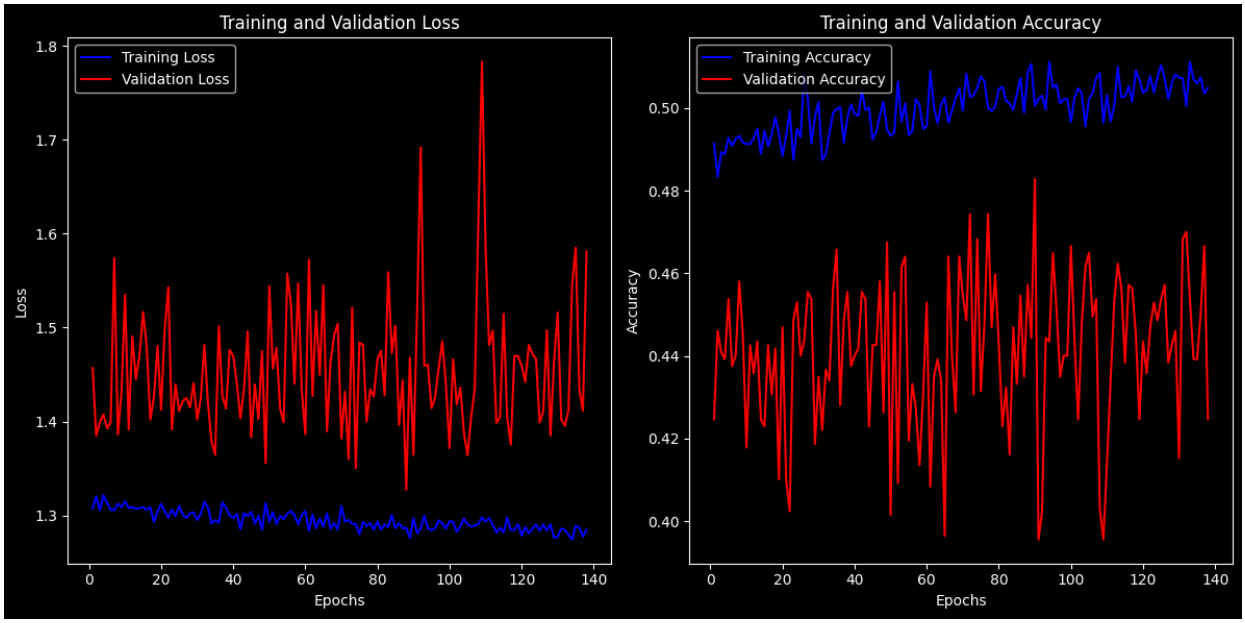
Model 2:

Architecture: Another variant of Bidirectional LSTM layers.
Dropout rates: 40% for LSTM layers, 60% for dense layers.
Optimizer: RMSprop with a learning rate of 0.001.
Loss function: Sparse categorical cross-entropy.

Visualisations-

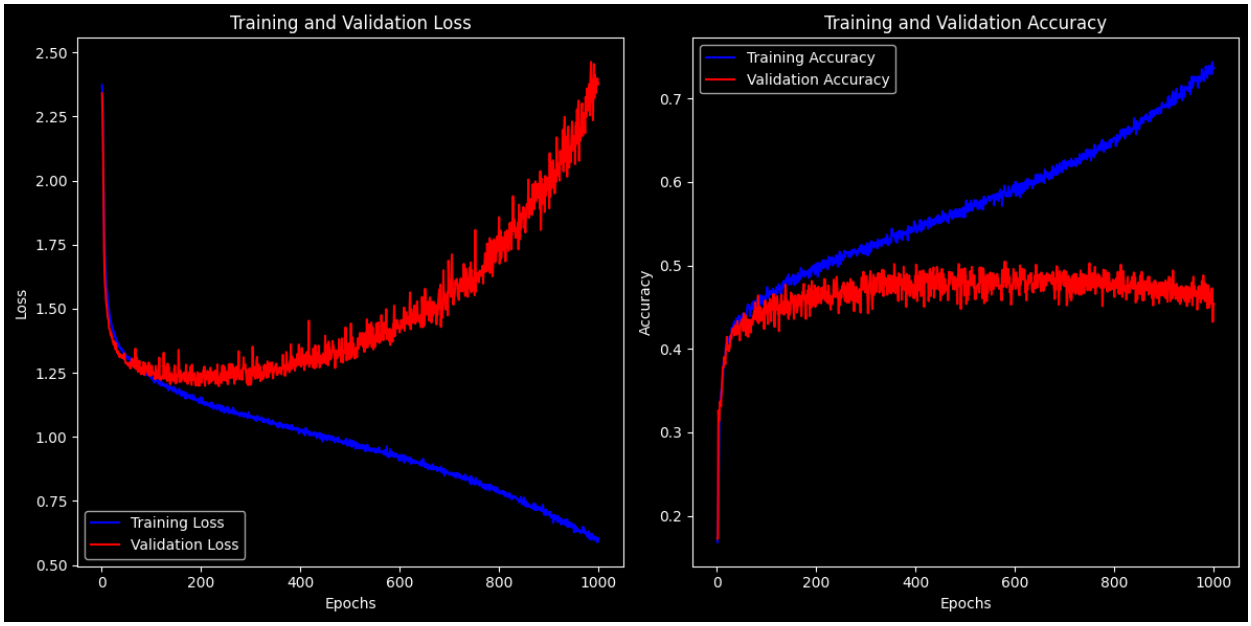
Now, let's visualise the training history for both models:

Model 1 Training History:



The plot shows the training and validation loss as well as the training and validation accuracy over epochs.
Observe any trends, overfitting, or convergence.

Model 2 Training History:



Similar to Model 1, this plot displays the training and validation metrics for Model 2. Compare the performance of both models.

Now that we've covered the technical aspects, let's discuss the practical implications. Your Streamlit app predicts essay grades based on content and language. Here are some considerations:

Explainability: Since you aim to provide explanations for the scores, consider incorporating interpretability techniques. For instance:

Attention Maps: Visualise which parts of the essay contribute most to the predicted grade. Word Importance: Identify influential words or phrases.

Feature Importance: Explore the impact of different features (e.g., essay length, vocabulary richness) on the score.

User Experience: Ensure that the app is user-friendly. Provide clear instructions, and consider adding features like highlighting key sentences or suggesting improvements. Feedback Loop: Collect feedback from users (teachers, students, etc.) to continuously improve the model. Consider integrating user feedback into the training process.

Future Implementation Plan: Roadmap for Further Development and Deployment

The development and deployment of an AI model for automated essay grading with explainability represent a significant step forward in educational technology. However, there are numerous opportunities for further enhancement and expansion to ensure the effectiveness, scalability, and sustainability of the project. In this detailed future implementation plan, we outline a comprehensive roadmap for further development and deployment, covering various aspects such as model improvement, scalability, user experience, and community engagement.

1. Model Improvement:

- Fine-tuning and Optimization: Continuously fine-tune and optimise the existing LSTM models to improve grading accuracy and efficiency. This includes experimenting with different architectures, hyperparameters, and regularisation techniques to achieve better performance.

- Integration of Advanced NLP Techniques: Explore the integration of advanced natural language processing (NLP) techniques such as transformer-based models (e.g., BERT, GPT) to capture more nuanced features and semantics of essays, leading to more accurate grading and explanation generation.

- **Multimodal Learning:** Investigate the incorporation of multimodal learning approaches that combine textual information with other modalities such as images, graphs, or audio to provide a more comprehensive assessment of essays, especially in subjects like art, science, and multimedia studies.

2. Scalability and Efficiency:

- **Parallelization and Distributed Computing:** Implement parallelization and distributed computing techniques to scale up the grading process, enabling the system to handle larger volumes of essays efficiently, particularly during peak times such as exam periods.

- **Optimised Data Pipeline:** Optimise the data preprocessing pipeline to minimise processing time and resource utilisation, allowing for faster turnaround times between essay submission and grading.

- **Cloud Deployment:** Deploy the grading system on cloud infrastructure (e.g., AWS, Google Cloud) to leverage auto-scaling capabilities and reduce operational overhead, ensuring high availability and reliability.

3. User Experience and Accessibility:

- **Interactive Feedback Mechanism:** Develop an interactive feedback mechanism that provides students with detailed insights into their essays' strengths and weaknesses, along with personalised suggestions for improvement. This could include highlighting specific areas for revision and providing annotated examples.

- **User-Friendly Interface:** Enhance the user interface of the grading application (e.g., Streamlit app) to be more intuitive, visually appealing, and accessible to users with diverse backgrounds and abilities.

- **Mobile Compatibility:** Ensure compatibility with mobile devices to allow students and educators to access the grading system on-the-go, facilitating seamless interaction and engagement.

4. Explainability and Interpretability:

- **Enhanced Explanation Generation:** Improve the quality and coherence of the explanations provided by the AI model by incorporating more sophisticated natural language generation techniques, such as controlled text generation and coherent text planning.

- **Interactive Explanations:** Enable users to interact with the generated explanations, allowing them to explore different aspects of the grading criteria and understand the reasoning behind specific scores in more detail.

- **Integration with Learning Analytics:** Integrate the essay grading system with learning analytics platforms to track students' progress over time, identify common areas of difficulty, and tailor feedback and support accordingly.

5. Community Engagement and Collaboration:

- **Open-Sourcing and Collaboration:** Consider open-sourcing the codebase and collaborating with other educational institutions, researchers, and developers to foster innovation, share best practices, and contribute to the advancement of automated essay grading technology.

- **User Feedback and Iterative Development:** Solicit feedback from users (educators, students, administrators) through surveys, interviews, and usability testing to identify pain points, prioritise feature requests, and guide iterative improvements.

- **Educational Partnerships:** Forge partnerships with educational organisations and stakeholders to pilot the grading system in real-world educational settings, gather data, and validate its effectiveness and impact on teaching and learning outcomes.

6. Ethical and Responsible AI:

- **Bias Detection and Mitigation**: Implement mechanisms to detect and mitigate biases in the grading system, ensuring fair and equitable assessment across diverse student populations and avoiding reinforcing existing societal inequalities.

- **Transparency and Accountability:** Provide transparency into the grading process and decision-making logic to build trust with users and ensure accountability for the system's outcomes. This includes documenting model training data, algorithms, and evaluation metrics.

- **Ethics Education:** Integrate ethics education modules into the grading system to raise awareness among users about the ethical implications of AI-driven assessment and promote responsible usage and interpretation of automated grading results.

Conclusion:

The future implementation plan outlined above provides a comprehensive roadmap for further development and deployment of the automated essay grading system with explainability. By focusing on model improvement, scalability, user experience, explainability, community engagement, and ethical considerations, the project aims to continuously evolve and meet the needs of educators, students, and educational institutions. Through ongoing collaboration, innovation, and commitment to responsible AI, the project has the potential to make a significant impact on the field of education, facilitating more efficient, transparent, and equitable assessment practices.