(Ghufron Wahyu Kurniawan / 413830003)

# Abstract

The rapid advancement of generative artificial intelligence has led to the proliferation of AI-generated texts that are often indistinguishable from human-authored content. Accurately differentiating between these two sources is increasingly critical for various applications, including information integrity, authorship verification, and digital forensics. This paper presents a deep learning-based classification framework that utilizes a Long Short-Term Memory (LSTM) network enhanced with an attention mechanism to effectively identify whether a given text is written by a human or generated by an AI model. To address the issue of class imbalance, the implement method incorporates a positional weighting strategy within the Binary Cross-Entropy with Logits Loss function. The model is trained and validated on a large-scale dataset comprising equal proportions of human-written and AI-generated texts. Experimental evaluations demonstrate that the attention mechanism significantly improves the model's ability to capture meaningful sequence information, leading to highly accurate classification performance. The proposed approach achieves a classification accuracy of 99.91% on the full dataset, indicating its effectiveness in discerning subtle linguistic differences between human and AI-generated texts.

**Keywords—**Text classification, LSTM, Attention mechanism, Deep learning, Binary classification, Human-AI text distinction, Token analysis, Neural networks.

(Ghufron Wahyu Kurniawan / 413830003)

## 1. Introduction

The rise of sophisticated AI models such as GPT-3 and other language models has made it increasingly difficult to distinguish between human-written and machine-generated text [1]. The ability to accurately classify such content is crucial in areas such as content moderation, automated content creation, and AI-based detection systems[1][2]. This work presents a model for binary text classification using a combination of Long Short-Term Memory (LSTM) and an attention mechanism to address the challenge of distinguishing human text from AI-generated content.

LSTM, a type of Recurrent Neural Network (RNN), is particularly effective for sequential data such as text, where it can capture long-range dependencies[3]. However, LSTM models struggle with handling sequences where only specific tokens are important, making it difficult to focus on significant words. This is where the attention mechanism comes in. The attention mechanism allows the model to assign varying levels of importance to different parts of the input sequence, improving the model's focus and interpretability[4].

This study presents a comprehensive methodology for constructing and evaluating a text classification model, emphasizing the synergistic integration of the LSTM network and attention mechanism to enhance classification accuracy.

## 2. Related Work

With the rise of advanced AI models, identifying AI-generated text has become increasingly important. Earlier rule-based techniques, which relied on spotting repetitive patterns or unnatural phrasing, often fail against sophisticated systems[5]. In recent years, Deep learning methods, particularly LSTM networks combining with another method, have shown improved performance by capturing contextual relationships and focusing on key parts of the text[6]. Additionally, Transformer-based models like BERT[7] have become state-of-the-art in text classification, leveraging self-attention to understand entire sequences at once, though they require more computational resources.

Some example in the classroom projects or teaching examples have used basic models to tell the LSTM using human and AI text dataset. These models are good for learning but not strong enough for real-world use. More recent work in the project example has explored comparation LSTM and CNN (Convolutional Neural Network) models to take advantage of both sequential information and local word patterns in the text.

**(Ghufron Wahyu Kurniawan / 413830003)**

Our work builds on these ideas by using an LSTM model with an attention mechanism to classify whether a text is human-written or AI-generated. We also use a special technique to handle imbalanced data, where there are more examples of one class than the other. This combination allows our model to be both accurate and reliable, making it a strong option for detecting AI-generated content.

## 3. Methodology

The proposed model for classifying human-written versus AI-generated text integrates a Long Short-Term Memory (LSTM) network with an attention mechanism. This hybrid architecture is designed to capture both the sequential dependencies of textual data and the importance of individual tokens in a sequence. The model is trained using the Binary Cross-Entropy with Logits Loss function, which allows for stable optimization by operating directly on raw logits, eliminating the need for an explicit sigmoid activation in the output layer.

### 3.1 Model Architecture

The architecture comprises the following components:

- Embedding Layer: This layer maps each token in the input sequence to a dense, high-dimensional vector space, capturing semantic relationships between words.
- LSTM Layer: The embedded sequence is passed through an LSTM network, which captures long-range dependencies and contextual information within the text.
- Attention Mechanism: To enhance interpretability and performance, an attention mechanism is applied to the LSTM outputs. It assigns dynamic weights to each token, allowing the model to emphasize more relevant parts of the sequence.
- Fully Connected Layer: A linear layer receives the attention-weighted context vector and outputs a scalar logit representing the classification score.
- Loss Function: The model is trained using the Binary Cross-Entropy with Logits Loss, which combines a sigmoid activation with binary cross-entropy in a numerically stable manner.

### 3.2 LSTM Layer

The LSTM layer is responsible for processing the sequential structure of text data. Given a sequence of embedded tokens, the LSTM produces hidden states for each time step, encapsulating contextual information. These hidden states are essential for understanding the relationships between words, especially in longer or more complex sentences. The LSTM's ability to retain both short- and long-term dependencies contributes significantly to the model's ability to differentiate between human-authored and machine-generated content.

### 3.3 Attention Mechanism

The attention mechanism enhances the model's focus on critical tokens within the input sequence. It computes a set of attention weights over the LSTM-generated hidden states using a learnable linear transformation followed by a softmax function. These weights are used to generate a context vector—a weighted sum of the hidden states—which captures the most salient features of the input text. This mechanism improves both the model's accuracy and interpretability, enabling it to prioritize meaningful parts of the sequence during classification.

### 3.4 Output Layer and Loss Function

The context vector obtained from the attention mechanism is passed to a fully connected linear layer that produces a single logit score. This raw output is not passed through a sigmoid activation; instead, the Binary Cross-Entropy with Logits Loss function is applied during training. This function internally applies the sigmoid transformation before computing the loss, offering both computational stability and simplicity in design.

### 3.5 Handling Variable-Length Sequences

To efficiently manage input sequences of varying lengths, the model utilizes packed sequences during training. Sequences are packed using the pack_padded_sequence function, which allows the LSTM to skip over padding tokens and focus only on meaningful content. After processing, the LSTM output is restored to its original format using pad_packed_sequence. This technique enhances computational efficiency and prevents the model from learning from irrelevant padding.

### 3.6 Regularization via Dropout

To mitigate the risk of overfitting, dropout regularization is applied to the context vector before it enters the fully connected layer. During training, dropout randomly zeroes a portion of the vector's components, encouraging the model to develop robust, generalizable features rather than relying on specific patterns.

**(Ghufron Wahyu Kurniawan / 413830003)**

## 4. Results and Discussion

Following the presentation of results, the discussion chapter interprets the findings and their relevance in the context of the AI-generated text. It explores the model’s performance, highlighting how the architecture contributed to the achieved accuracy.

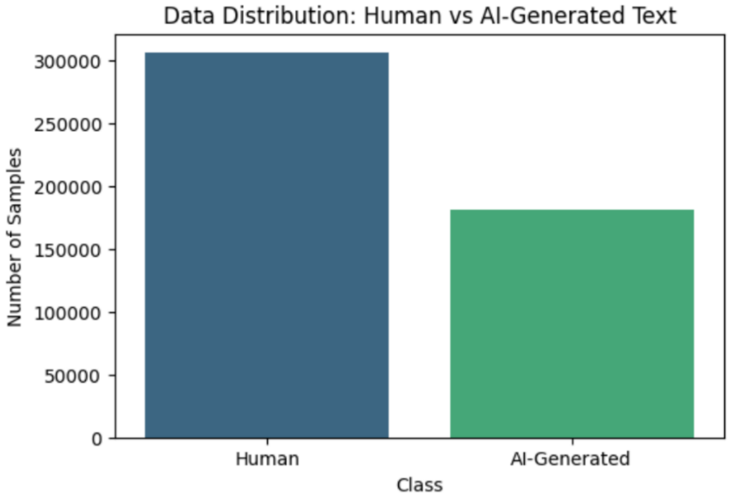


Figure 1. Data Distribution of Human-Written vs AI-Generated Texts by Class.

### 4.1. Discussion on Data Distribution of Human-Written vs AI-Generated Texts by Class

Figure 1 illustrates a histogram of token counts for both human-written and AI-generated texts. This visualization provides insights into the structural characteristics of the dataset, revealing notable differences in length variability between the two text types. AI-generated samples exhibit a more consistent token length distribution, which is a consequence of their algorithmic generation. In contrast, human-written texts demonstrate greater variability due to the diversity of individual writing styles.

In the histogram, the x-axis represents the number of tokens per sample, while the y-axis indicates the frequency of samples within each token length range. This distributional information is crucial for understanding how text length may affect classification performance and can inform preprocessing strategies such as padding or truncation. It also guides the design of models to ensure robustness across varying text lengths.

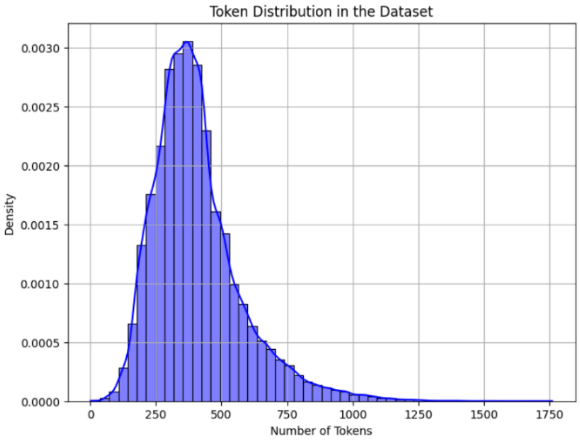


Figure 2. Token Distribution Across Human-Written and AI-Generated Texts.

### 4.2. Discussion on Overall Token Distribution in Human-Written and AI-Generated Texts Dataset

Figure 2 presents the global token distribution across the dataset. The x-axis denotes the number of tokens per sample, and the y-axis indicates the density of samples at each length. The distribution follows a unimodal, positively skewed pattern, with most samples concentrated around 250 tokens. This suggests that the dataset predominantly consists of medium-length texts, with relatively fewer instances of very short or very long samples. This characteristic supports the use of models optimized for mid-range sequences, with flexibility to adapt to outliers.

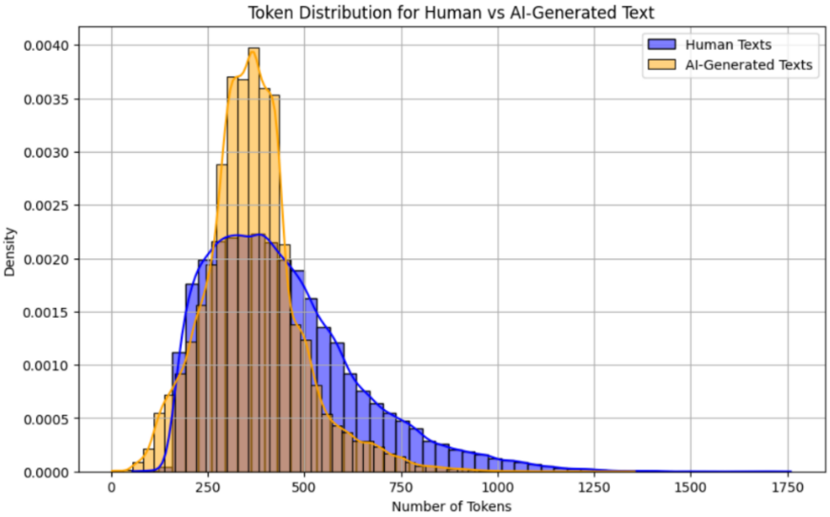


Figure 3 Token distribution for Human vs AI-Generated text by number of token

**(Ghufron Wahyu Kurniawan / 413830003)**

### 4.3. Comparative Token Length Analysis between Human-Written and AI-Generated Texts

Figure 3 offers a comparative visualization of token distributions for human-written (blue) and AI-generated (orange) samples. Human-written texts exhibit a broader and more symmetrical distribution centered near 250 tokens, whereas AI-generated texts display a narrower, more peaked distribution with a slight shift toward longer token lengths. This reflects the structured and often verbose nature of AI-generated content, which tends to maintain a consistent style and length. The overlap in distributions highlights the challenge in classification, while the subtle shifts point to exploitable stylistic and structural differences.

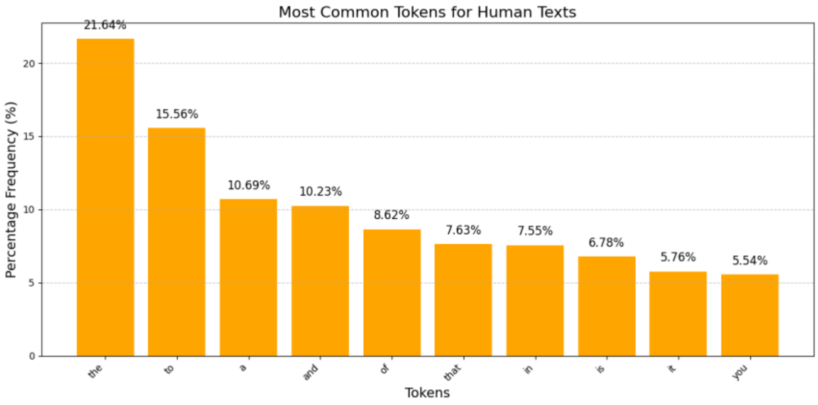


Figure 4 The Most Common Tokens for Human Text

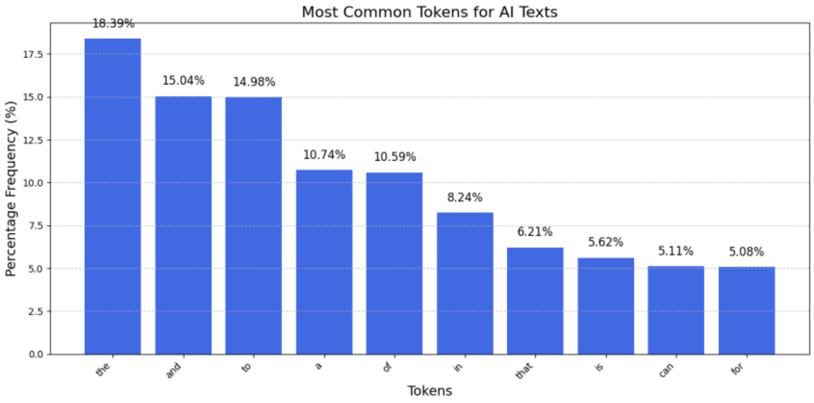


Figure 5 The Most Common Tokens for AI-Generated text

### 4.4. Token Frequency Analysis

Figures 4 and 5 present the token frequency distributions for human-written and AI-generated texts, respectively. Human-authored texts (Figure 4) show a high frequency of common function words, with “the” (21.64%), “to” (15.56%), “a” (10.69%), and “and” (10.23%) being the most prevalent. This reflects natural language tendencies, where determiners, prepositions, and conjunctions facilitate narrative cohesion and context.

In contrast, AI-generated texts (Figure 5) also include frequent function words—“the” (18.39%), “and” (15.04%), and “to” (14.98%)—but with more balanced proportions. This more uniform distribution suggests a formulaic generation style, optimized for coherence and fluency. Interestingly, the token “you” appears only in human-written texts (5.54%), indicating a more personal or conversational tone. Conversely, “can” (5.11%) and “for” (5.08%) are more prominent in AI-generated texts, reflecting a tendency toward instructive or explanatory output.

Differences in frequencies of tokens such as “that,” “in,” and “is” further underscore stylistic divergences between the two text types. These linguistic patterns provide valuable cues for classification models and offer deeper insights into the generative behavior of language models.

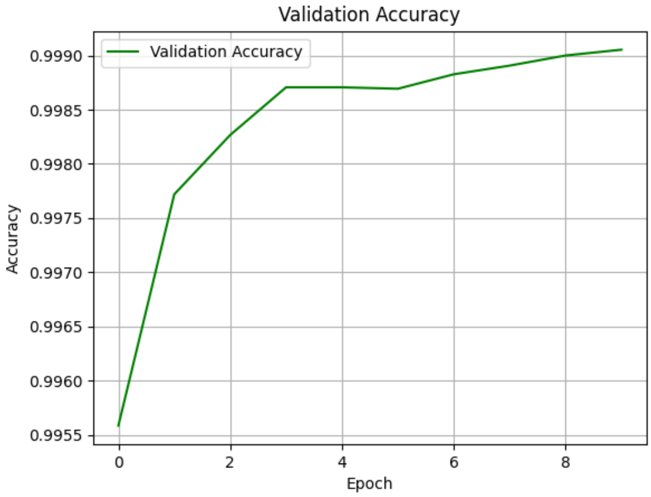


Figure 6 Validation Accuracy of proposed model

### 4.5. Model Performance Evaluation

The proposed model was trained for 10 epochs, showing strong and consistent learning behavior. As shown in Figure 6, validation accuracy improved steadily, reaching 99.91% by the final epoch. This indicates that the model successfully generalized to unseen data and learned meaningful representations from the training set.

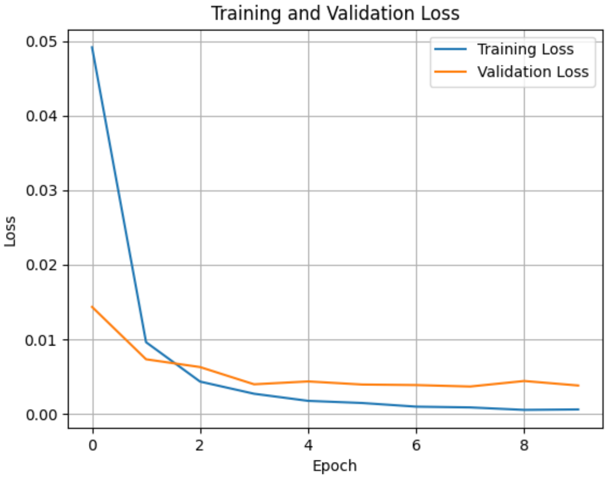
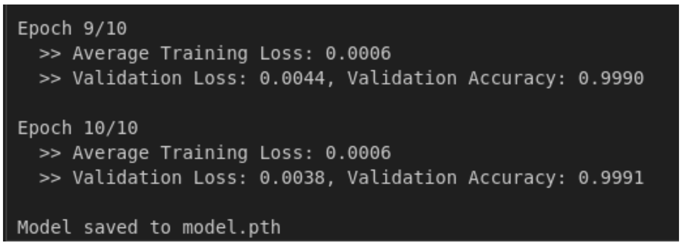


Figure 7 Training and Validation Loss of proposed model each epoch.

Figure 7 depicts the training and validation loss curves, both of which show substantial declines during the early epochs. The training loss dropped sharply and gradually approached near-zero values, while the validation loss decreased and stabilized at a low level without significant fluctuation. This close alignment between the two curves suggests that the model avoided overfitting and maintained robust generalization.



```
Epoch 9/10
  >> Average Training Loss: 0.0006
  >> Validation Loss: 0.0044, Validation Accuracy: 0.9990

Epoch 10/10
  >> Average Training Loss: 0.0006
  >> Validation Loss: 0.0038, Validation Accuracy: 0.9991

Model saved to model.pth
```

Figure 8 Training and Validation Losses Over Epochs

Further confirmation is provided in Figure 8, which summarizes average training and validation losses per epoch. These logs support the conclusion that the model progressively minimized loss while maintaining validation stability. The final model checkpoint was saved at epoch 10, corresponding to the peak validation accuracy of 99.91%, confirming this as the optimal stopping point.

## 5. Conclussion

In this study, a hybrid LSTM-based model with an attention mechanism was developed to classify human-written and AI-generated texts. The methodology employed a well-structured model architecture that incorporated an embedding layer, LSTM layer, attention mechanism, and a fully connected output layer, all designed to enhance the model's ability to learn from text sequences of varying lengths. The model was trained using Binary Cross-Entropy with Logits Loss, which facilitated stable optimization during training. Key elements of the design included the handling of variable-length sequences through packed sequences, and regularization using dropout to prevent overfitting.

The dataset, consisting of over 300,000 samples with a substantial portion of AI-generated content, was analyzed to explore token length distribution, token frequency, and structural differences between the two classes. The analysis revealed distinct differences in the token distributions between human-written and AI-generated texts, with human-written texts exhibiting more variability and AI-generated texts showing more consistent length patterns. Token frequency analysis further highlighted stylistic divergences, with certain function words being more prevalent in each class.

The model achieved exceptional performance, with validation accuracy reaching 99.91% after 10 epochs, demonstrating the effectiveness of the hybrid LSTM-attention model in distinguishing between human and AI-generated texts. The training and validation loss curves supported this result, showing steady improvement and confirming the model’s robustness against overfitting. This work contributes valuable insights into AI content detection, with promising potential for further advancements in the field.

**(Ghufron Wahyu Kurniawan / 413830003)**

## References

[1] C. Clark, J. K. Kummerfeld, and Y. Choi, "Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text," *arXiv preprint arXiv:2107.01294*, 2021.

[2] X. Yang, W. Cheng, Y. Wu, L. Petzold, W. Y. Wang, and H. Chen, "DNA-GPT: Divergent N-Gram Analysis for Training-Free Detection of GPT-Generated Text," arXiv preprint arXiv:2305.17359, 2023.

[3] J. Zhang, L. Zhao, and Z. Li, "A deep learning model based on LSTM for text classification," IEEE Access, vol. 9, pp. 53453-53462, 2021.

[4] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[5] K. Zellers, A. Holtzman, A. L. Lu, F. Liu, and Y. Choi, "Defending against neural fake news," Proceedings of the 2019 Conference on Neural Information Processing Systems (NeurIPS), 2019.

[6] M. Lin, Q. Liu, H. Wei, and Z. Wei, "A hybrid deep learning model for text classification," IEEE Access, vol. 7, pp. 109412-109423, 2019.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. of the 2019 North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019), Minneapolis, MN, USA, 2019, pp. 4171-4186.