Team EssayDetect at GenAI Detection Task 2: Guardians of Academic Integrity: Multilingual Detection of AI-Generated Essays

Anonymous ACL submission

Abstract

Detecting AI text in the field of Academic is become very prominent.This paper presents a solution for Task 2: AI vs. Human – Academic Essay Authenticity Challenge in the COLING 2025 DAIGenC Workshop. The rise of large language models (LLMs) like ChatGPT has posed significant challenges to academic integrity, particularly in detecting AI-generated essays. We propose a fusion model that combines pre-trained language model embeddings with stylometric and linguistic features to improve classification accuracy. Our approach, tested on both English and Arabic, utilizes adaptive training and attention mechanisms to enhance F1 scores, address class imbalance, and capture linguistic nuances across languages. This work advances multilingual solutions for detecting AI-generated text in academia.

1 Introduction

With the increasing prevalence of large language models (LLMs), detecting AI-generated text in academic settings, such as essays, has become crucial. AI tools like ChatGPT are gaining widespread attention and raising concerns about academic integrity. Social media has recently seen a surge in discussions about LLM releases and their diverse applications, including language translation, summarization, question answering, and text generation. Many posts advocate using AI-generated academic content, such as composing essays and crafting content-specific questions. However, using AI-generated content in academic contexts poses challenges related to academic integrity, plagiarism, and associated consequences (Liao, 2020). Much research is underway to address the challenge of detecting AI-generated content. Researchers are developing various methods and models to distinguish AI-produced text from human-authored content.

To tackle this issue, The GenAI Content Detection Task 2, *"AI vs. Human – Academic Essay Authenticity Challenge,"* aims to identify machine-generated essays to safeguard academic integrity and prevent misuse of LLMs in education. The input consists of essays authored by both native and non-native speakers, along with texts generated by various LLMs. The task, framed as follows—*"Given an essay, identify whether it is generated by a machine or authored by a human"*—is a binary classification challenge divided into two sub-tasks: Subtask A for English essays and Subtask B for Arabic.

We evaluated multiple approaches and ultimately focused on two strategies. While only one approach was submitted for leaderboard consideration, the other offers valuable insights. We developed several feature-based models to address this task, detailed in Section 3. We fine-tuned the Pre-Trained Language Model (PLM), which initially performed poorly and exhibited a bias toward the majority class in the dataset. We integrated linguistic and stylistic features to address this, resulting in an improved overall F1 score within the fusion model. We mainly focus on three main problems: capture the dependencies within feature and higher discriminant features, address the class imbalance, and Improve training to focus on maintaining basic linguistic representations in the lower layers while allowing the higher layers to capture task-specific (essay) stylistic differences.

2 Background

Over the last few years, numerous approaches have been proposed to tackle the task of AI-generated text detection.Detecting machine-generated text is primarily formulated as a binary classification task (Zellers et al., 2019; Gehrmann et al., 2019; Ippolito et al., 2019), naively distinguishing between human-written and machine-generated text. In gen-

eral, there are three main approaches: the supervised methods (Wang et al., 2023; Uchendu et al., 2021; Zellers et al., 2019; Zhong et al., 2020; Liu et al., 2023, 2022), the unsupervised ones, such as zero-shot methods (Solaiman et al., 2019; Ippolito et al., 2019; Mitchell et al., 2023; Su et al., 2023; Hans et al.; Shijaku and Canhasi, 2023) and Adversarial measures on detection accuracy (Susnjak and McIntosh, 2024; Liang et al., 2023), especially within the education domain. For example, (Antoun et al., 2023) evaluates the robustness of detectors against character-level perturbations or misspelled words, focusing on French as a case study. (Krishna et al., 2024) train a generative model (DIPPER) to paraphrase paragraphs to evade detection. Although supervised approaches yield relatively better results, they are susceptible to overfitting (Mitchell et al., 2023; Su et al., 2023).

There are some techniques like feature-based, fusion, and ensemble methods, such as word count, vocabulary richness, and readability concatenated ML, Neural based or finetuned (Solaiman et al., 2019; Kumarage et al., 2023; Shah et al., 2023; Nguyen-Son et al., 2017; Mindner et al., 2023; Kumarage and Liu, 2023).

3 Methodology

We use a fusion model approach: first, we extract all discriminative features that differentiate AI-generated text from human-written text. These features are then concatenated with those from a pre-trained language model (PLM) and trained together in a feed-forward model.

3.1 Stylometric Features

The stylometric features aim to capture different stylistic signals within a given text. These features are grouped into three categories: **Phraseology**, which quantifies how the author organizes words and phrases (e.g., average word count, sentence count); **Lexical Diversity**, which measures how varied the author's vocabulary is (e.g., lexical richness, readability scores); and **Syntactic Diversity**, which assesses how the author structures sentences and conveys emotions (e.g., sentiment scores). Table 1 summarizes the complete set of features used in each of these three categories.

3.2 Model

Our model for AI-generated text detection combines pre-trained language model *(PLM)* embeddings with stylometric features, leveraging attention mechanisms and adaptive training to enhance classification performance. For each input instance, we first extract the stylometric features and apply *LIME* (Local Interpretable Model-agnostic Explanations) to select the most distinguishing feature as a vector $\mathbf{s}_K \in \mathbb{R}^K$, where K is the number of stylometric features. These features help distinguish between human and AI-generated texts. In parallel, we obtain the CLS token embedding from the final hidden layer of the PLM, denoted as $\mathbf{h}^e_{\text{CLS}} \in \mathbb{R}^e$, where e is the embedding size of the model's output. This embedding captures the semantic meaning of the entire input text.

To capture the dependencies within the stylometric features, we apply a *self-attention* mechanism over the stylometric features, producing an attention-weighted vector $\mathbf{s}^{\text{att}}_K$, which emphasizes the most relevant stylometric attributes for the task. The attention mechanism can be mathematically defined as Eq.1.

$$\mathbf{s}^{\text{att}}_K = \text{Attention}(\mathbf{s}_K) \tag{1}$$

where the attention function assigns weights to each stylometric feature based on its relevance to the classification task.

Next, we concatenate the attention-weighted stylometric vector $\mathbf{s}^{\text{att}}_K$ with the CLS token embedding $\mathbf{h}^e_{\text{CLS}}$ to create a combined feature vector $\mathbf{f}_{\text{concat}}$.

$$\mathbf{f}_{\text{concat}} = [\mathbf{s}^{\text{att}}_K; \mathbf{h}^e_{\text{CLS}}] \in \mathbb{R}^{K+e} \tag{2}$$

This concatenated vector is then passed through a reduce network, which consists of residual fully connected layers that map the concatenated vector to a reduced representation $\mathbf{r} \in \mathbb{R}^d$, where d is the dimensionality of the reduced vector. This is done by applying a learned function $\mathbf{r} = \mathbf{f}^\theta_{\text{concat}}$, where θ represents the parameters of the network.

Finally, the reduced representation $\mathbf{r}$ is passed through a classification network consisting of additional fully connected layers followed by a softmax activation, producing the class probabilities $p_\theta(y|\mathbf{r})$ for the input text, where $y \in \{0, 1\}$ is the label, with 0 indicating "human-written" and 1 indicating "AI-generated." The softmax function is defined in Eq. 3.

$$p_\theta(y|\mathbf{r}) = \frac{\exp(\mathbf{W}_y^T \mathbf{r} + b_y)}{\sum_{y'} \exp(\mathbf{W}_{y'}^T \mathbf{r} + b_{y'})} \tag{3}$$

where $\mathbf{W}_y$ and b_y are the weight vector and bias for class y, respectively.

Stylometry Analysis	Feature Sets
Phraseology	Word count, sentence count, paragraph count, and mean and standard deviation of word count per sentence, word count per paragraph, Total punctuation count, Exclamation count and sentence count per paragraph
Lexical Diversity	syllables count, comma count, stopwords count, unique words count, Lexical Diversity, Type token ratio, Flesch reading ease, Flesch Kincaid grade and Gunning fog
Syntactic Diversity	Sentiment polarity, Sentiment subjectivity, Proportion of nouns, Proportion of verbs, Proportion of adjectives and Proportion of adverbs

Table 1: Different stylometric feature categories and corresponding feature sets

To address class imbalance, we use focal loss, which modifies the standard cross-entropy loss by focusing more on difficult-to-classify examples. The focal loss for an input $\mathbf{r}$ and label y is given by Eq. 4.

$$\mathcal{L}_{\text{focal}} = -\alpha(1 - p_\theta(y|\mathbf{r}))^\gamma \log(p_\theta(y|\mathbf{r})) \quad (4)$$

where α is a balancing factor that adjusts the importance of the classes, and γ is a focusing parameter that helps to down-weight easy examples. The value of γ is typically set between 0 and 5, with higher values making the model focus more on hard-to-classify instances. Specifically, γ controls the rate at which the modulating factor $(1-p_\theta(y|\mathbf{r}))^\gamma$ decreases the loss for well-classified instances.

The fusion model leverages stylometric features $\mathbf{s}_K$ and semantic embeddings $\mathbf{h}^e_{\text{CLS}}$ to classify text as human-written or AI-generated. To improve performance, we apply layer-wise freezing during fine-tuning. Let the layers of the PLM be represented as $L_1, L_2, \ldots, L_n$, where L_1 is the lowest layer and L_n is the highest. We freeze the parameters $\theta_{L_1}, \ldots, \theta_{L_k}$ for lower layers and update $\theta_{L_{k+1}}, \ldots, \theta_{L_n}$ for higher layers, which can be formalized as Eq. 5.

$$L_{\text{fine-tune}} = \sum_{i=k+1}^{n} L(\theta_{L_i}). \quad (5)$$

This helps the model maintain basic linguistic representations in the lower layers while allowing the higher layers to capture task-specific stylistic differences. The model is trained with cross-entropy or focal loss and optimized using back-propagation, focusing on the relevant features for AI-generated text detection.

4 Experiments

4.1 Data

Shared task organizers have been used for task A, English essay, and sub-task B, Arabic essay, with labels "ai" and "human." There are three datasets: one for Training and development data with labels; for the development phase, they provided development data without labels; and for the Evaluation phase, they provided testing data without labels for both tasks. All descriptions mentioned in Table 2

Data	Train		Dev		Dev	Test
	AI	Human	AI	Human	Total	Total
English	1467	629	391	1235	567	1130
Arabic	925	1145	299	182	293	886

Table 2: Dataset distribution across train, dev, dev without label and test without label set.

4.2 Experimental setup

Table 6 presents the details of hyperparameters used to produce the results presented in this paper. In this study incorporate Macro F1 (Classwise Score) performance evaluation metrics.Further details of experimental setup in presented in section A.1.

4.3 Feature and Model Selection

To improve model interpretability, we use **LIME** (Local Interpretable Model-agnostic Explanations) for feature selection, helping identify the most influential features for detecting AI-generated text as detailed in A.1.1. For subtask A (English essays), Table 7, including the maximum, minimum, and average feature scores for AI and human text, lists 21 linguistic and stylometric features, with LIME highlighting the top 12 most discriminative

features, as shown in Fig. **??**. However, certain features, such as part-of-speech (POS) tags, are not applicable for subtask B (Arabic essays). Therefore, For subtask B (Arabic essays), Table 8 considers 15 features, with LIME selecting the ten most discriminative ones, shown in Fig. 1.

In summary, the model relies on linguistic features like word length, sentence length, punctuation frequency and human-like features such as vocabulary diversity and readability. Higher values in these "human-like" features make the model classify the text as human-written with high confidence..

For this experiment, we consider pretrained language models such as *RoBERTa* (Liu, 2019), *BERT* (Devlin, 2018), *DeBERTa* (He et al., 2020), and *DistilBERT* (Sanh, 2019) for Subtask A, which focuses on English essays. For Subtask B, we use multilingual pretrained language models, including *XLM-RoBERTa* (Wiciaputra et al., 2021) and *AraBERT* (Antoun et al., 2020), both of which are Transformer-based models designed for Arabic language understanding.

5 Results

In this section, we analyze the results of Subtask A and Subtask B. Subtask A focused on feature extraction and evaluating different models, while Subtask B also involved feature extraction with a different set of models. Both subtasks aimed to assess the increase the classwise F1 score to evaluate model performance.

5.1 Subtask A

Table 3 shows that adding embeddings and stylometric diversity significantly boosts model performance. The baseline model scores 0.478, while *BERT-base-uncased* improves from 0.567 to 0.818 with features. *DistilBERT-base-uncased* scores 0.931, but *DeBERTa-base* achieves the highest score of 0.978 with added features, highlighting the strong impact of these enhancements for Task A in English.

5.2 Subtask B

Table 4 presents the F1 scores for various models on the test data for Task A in Arabic, both with and without additional features. The results show that *AraBERT v02* achieves the highest score of 0.9214 without any features. However, when features are included, *AraBERT-base* outperforms all models with an F1 score of 0.9429, followed closely by *XLM-RoBERTa-base* at 0.9414. The baseline model achieves the lowest score of 0.4605, demonstrating the significant impact of adding features on model performance

Model	Feature	F1
Baseline	-	0.478
RoBERTa-base	-	0.462
BERT-base-uncased	-	0.567
DeBERTa-base	-	0.617
BERT-base-uncased	yes	0.818
RoBERTa-base	yes	0.796
DistilBERT-base-uncased	yes	0.931
DeBERTa-base	yes	**0.978**

Table 3: Model Performance Test Data with and without Features for Task A English

Model	Feature	F1
Baseline	-	0.4605
XLM-RoBERTa-base	-	0.9188
AraBERT v02	-	0.9214
XLM-RoBERTa-base	yes	0.9414
AraBERT-base	yes	**0.9429**

Table 4: Model Performance Test Data with and without Features for Task A Arabic

6 Analysis

Our main configuration achieved strong results in both English and Arabic tasks mention in Table 5. These outcomes highlight the effectiveness of our approach, where embeddings and stylometric diversity significantly enhance performance across both tasks.

Task	Acc.	P	R	F1	Rank
A-English	0.978	0.968	0.984	0.975	10
B-Arabic	0.942	0.949	0.919	0.932	13

Table 5: Performance Metrics for my in English and Arabic

7 Conclusions

Our contributions in this work are threefold: (1) capturing dependencies within features and identifying highly discriminative ones, (2) addressing

class imbalance, and (3) enhancing training by preserving core linguistic representations in the lower layers while allowing higher layers to capture task-specific stylistic differences in essays. These strategies have significantly improved model performance, demonstrating that targeted feature selection, class balancing, and layer-wise adjustments effectively capture stylistic nuances. Future work may refine these techniques to further enhance model accuracy.

8 Limitations

Our work is limited to the English and arabic language only as we opted to participate in a single Subtask of Coling Workshop. In addition, this work is only limited to the essay and LLMs included in the shared task data, therefore, the generalizability of our approach beyond these essay and LLMs will need to be verified in future experiment

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is chatgpt that easy to detect? In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1: travaux de recherche originaux–articles longs*, pages 14–27.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

A Hans, A Schwarzschild, V Cherepanova, H Kazemi, A Saha, M Goldblum, J Geiping, and T Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. *URL: https://arxiv.org/abs/2401.12070*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*.

Tharindu Kumarage and Huan Liu. 2023. Neural authorship attribution: Stylometric analysis on large language models. In *2023 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 51–54. IEEE Computer Society.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

S Matthew Liao. 2020. *Ethics of artificial intelligence*. Oxford University Press.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2022. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Lorenz Mindner, Tim Schlippe, and Kristina Schaaff. 2023. Classification of human-and ai-generated texts: Investigating features for chatgpt. In *International Conference on Artificial Intelligence in Education Technology*, pages 152–170. Springer.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Hoang-Quoc Nguyen-Son, Ngoc-Dung T Tieu, Huy H Nguyen, Junichi Yamagishi, and Isao Echi Zen. 2017. Identifying computer-generated text using statistical analysis. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1504–1511. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style,

high-performance deep learning library. *Advances in neural information processing systems*, 32.

V Sanh. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Aditya Shah, Prateek Ranka, Urmi Dedhia, Shruti Prasad, Siddhi Muni, and Kiran Bhowmick. 2023. Detecting and unmasking ai-generated texts through explainable artificial intelligence using stylistic features. *International Journal of Advanced Computer Science and Applications*, 14(10).

Rexhep Shijaku and Ercan Canhasi. 2023. Chatgpt generated text detection. *Publisher: Unpublished*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Teo Susnjak and Timothy R McIntosh. 2024. Chatgpt: The end of online exam integrity? *Education Sciences*, 14(6):656.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. 2023. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt. *ArXiv*, abs/2306.07401.

Yakobus Keenan Wiciaputra, Julio Christian Young, and Andre Rusli. 2021. Bilingual text classification in english and indonesian via transfer learning using xlm-roberta. *International Journal of Advances in Soft Computing & Its Applications*, 13(3).

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. *arXiv preprint arXiv:2010.07475*.

A Example Appendix

A.1 Details of Experimental Setups, Experimental Datasets and Hyperparameters

In our experimental setup, we use two different configurations. In the first setup, we fine-tune the Pre-Trained Language Model (PLM) independently for each subtask, adjusting the model to the specific characteristics of each task. The PLM is fine-tuned over 5 epochs with a learning rate of 2×10^{-5}, using the Adam optimizer and L2 regularization with a weight decay of 0.01. For the second setup, the PLM serves as the source of embeddings. A feed-forward neural network (FFNN) with two hidden layers takes these embeddings as input. Each hidden layer utilizes ReLU activation, batch normalization, and dropout with a rate of 0.5. The FFNN is trained with a learning rate of 2×10^{-5}, L2 regularization of 0.01, and early stopping after 25 epochs to avoid overfitting. To address class imbalance, we use focal loss, which modifies the standard cross-entropy loss by focusing more on difficult-to-classify examples. This allows the model to give more attention to the minority class. The entire training and evaluation process is implemented in PyTorch (Paszke et al., 2019), utilizing its capabilities for model optimization, training stability, and efficient handling of large-scale datasets.

Hyperparameter	Setup: Fine-tuning PLM
Epochs	10-250
Batch Size	5
Learning Rate	2×10^{-5}
Optimizer	Adam
L2 Regularization	Weight decay: 0.01
Loss Function	Focal Loss

Table 6: Hyperparameter settings for Setup 1: Fine-tuning PLM.

A.1.1 Detail of LIME using Feature Selection

To gain better insight into which features are most influential in detecting AI-generated text, we apply LIME (Local Interpretable Model-agnostic Explanations). LIME helps us understand the decision-making process of the black-box model by approximating it with a locally interpretable surrogate model, such as a linear regression or decision tree, around each individual prediction.

For each instance, LIME perturbs the input data (in our case, the feature vector $\mathbf{f_{concat}}$) and observes how the model's prediction changes. The goal is to identify which features, when altered, have the largest impact on the prediction. LIME assigns a weight to each feature, indicating its importance for a particular decision.

This process helps in selecting the most relevant features for the model by analyzing their contribution to the model's output. Features with higher importance scores are considered more influential, while less impactful features may be discarded. This feature selection process can improve model interpretability and potentially reduce overfitting by focusing on the most significant features.

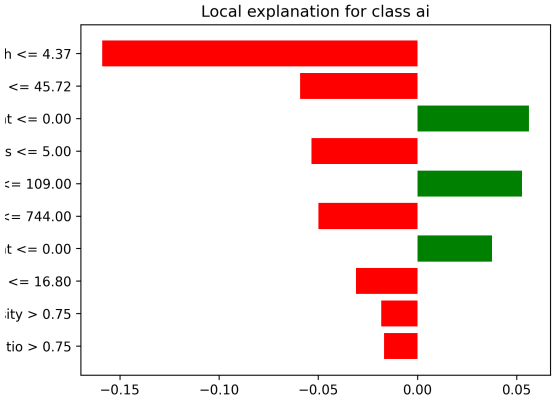

Figure 1: LIME Explanation for Subtask B

The implementation of LIME involves the following steps:

- Generate a surrogate interpretable model (e.g., decision tree) for each prediction.
- Perturb the input features and observe the changes in model prediction.
- Rank features based on their contribution to the prediction.
- Select the most influential features for further model refinement.

By using LIME, we ensure that the features contributing most to distinguishing human-written from AI-generated essays are given higher importance during training, potentially improving the model's performance and explainability.

A.2 Feature description

B Stylometry Analysis Feature Sets

B.1 Phraseology

- **Word Count (WC):** The total number of words in the text:

$$\text{WC} = \sum_{i=1}^{N} w_i$$

where w_i represents each word and N is the total number of words.

- **Sentence Count (SC):** The total number of sentences in the text.
- **Punctuation Count (TPC):** The total number of punctuation marks in the text.
- **Exclamation Count (EC):** The total number of exclamation marks in the text.

The phraseology features analyze the structure of the text, such as word, sentence, and paragraph counts, along with punctuation-related features like exclamation counts. These features help in understanding how the text is organized and how frequent punctuation marks are used.

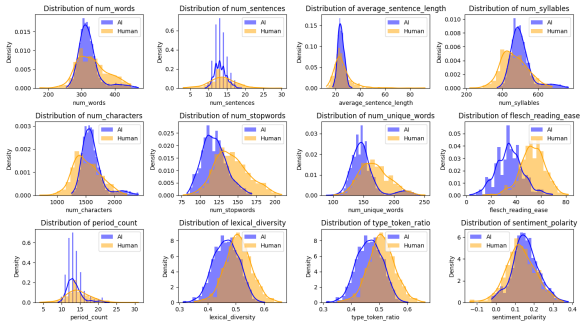

Figure 2: Distribution of features for AI and Human labels.

B.2 Lexical Diversity

- **Lexical Diversity (LD):** A measure of lexical variety:

$$\text{LD} = \frac{UWC}{WC}$$

where UWC is the number of unique words and WC is the total word count.

- **Type-Token Ratio (TTR):** Another measure of lexical diversity:

$$\text{TTR} = \frac{UWC}{WC}$$

- **Flesch Reading Ease (FRE):** A readability test:

$$\text{FRE} = 206.835 - 1.015 \times \left(\frac{WC}{SC}\right) - 84.6 \times \left(\frac{SC}{\text{Syllables}}\right)$$

Index	Feature	Max		Min		Avg	
		AI	Human	AI	Human	AI	Human
1	#words	471.0	254.0	321.37	449.0	174.0	332.21
2	#sentences	19.0	9.0	13.22	30.0	3.0	13.68
3	avg. sentence length	33.22	17.78	24.48	92.0	12.74	26.16
4	#syllables	770.0	372.0	504.96	680.0	218.0	467.32
5	#characters	2412.0	1254.0	1609.69	2212.0	703.0	1518.70
6	#stopwords	169.0	82.0	118.84	209.0	77.0	141.71
7	#unique words	243.0	89.0	149.73	251.0	101.0	168.19
8	flesch reading ease	69.31	2.85	34.67	81.93	13.35	53.25
9	flesch kincaid grade	17.8	8.3	13.71	25.6	5.5	11.17
10	gunning fog	18.68	9.72	13.89	26.74	6.72	12.52
11	#comma	42.0	10.0	22.42	38.0	1.0	15.72
12	#period	23.0	9.0	13.52	31.0	4.0	14.58
13	#exclamation	0.0	0.0	0.0	3.0	0.0	0.03
14	type token ratio	0.602	0.312	0.466	0.663	0.352	0.508
15	lexical diversity	0.602	0.312	0.466	0.663	0.352	0.508
16	sentiment polarity	0.380	-0.023	0.155	0.355	-0.138	0.130
17	sentiment subjectivity	0.709	0.208	0.445	0.722	0.284	0.472
18	pos proportion noun	0.330	0.171	0.255	0.322	0.144	0.230
19	pos proportion verb	0.180	0.064	0.113	0.193	0.067	0.119
20	pos proportion adj	0.179	0.049	0.112	0.176	0.038	0.089
21	pos proportion adv	0.088	0.006	0.040	0.098	0.011	0.048

Table 7: Linguistic and Stylometric Features Comparison in English Essay

where WC is word count, SC is sentence count, and Syllables is the total syllables count.

- **Flesch-Kincaid Grade (FKG):** A readability metric indicating the U.S. school grade level required to understand the text:

$$\text{FKG} = 0.39 \times \left(\frac{WC}{SC}\right) + 11.8 \times \left(\frac{\text{Syllables}}{WC}\right) - 15.59$$

- **Gunning Fog Index (GFI):** A readability test estimating the years of formal education required to understand the text:

$$\text{GFI} = 0.4 \times \left(\frac{WC}{SC} + 100 \times \frac{\text{Complex Words}}{WC}\right)$$

where complex words are those with three or more syllables.

Lexical diversity features provide insight into the richness of vocabulary used in the text. Measures like the Type-Token Ratio (TTR) and Lexical Diversity (LD) indicate how varied the vocabulary is. Readability scores like the Flesch Reading Ease (FRE) help assess how easy it is to read the text.

B.3 Syntactic Diversity

- **Proportion of Nouns (PN):** The proportion of nouns in the text relative to the total number of words:

$$\text{PN} = \frac{\text{Number of Nouns}}{WC}$$

- **Sentiment Polarity (SP):** A measure of the emotional tone of the text, ranging from -1 (negative) to 1 (positive).

- **Sentiment Subjectivity (SS):** A measure of how subjective or opinion-based the text is, usually ranging from 0 (objective) to 1 (subjective). This score is calculated by analyzing the presence of subjective words.

Syntactic diversity features examine sentence structure and sentiment. The proportion of different word types (e.g., nouns, verbs) helps determine the syntactic complexity of the text. Sentiment polarity quantifies the overall sentiment (positive or negative) of the text.

Index	Feature	Max		Min		Avg	
		AI	Human	AI	Human	AI	Human
1	num_words	1555	664	45	54	215.11	251.17
2	num_sentences	223	38	2	1	13.34	7.13
3	average_sentence_length	453	524	5.94	5.60	17.39	73.09
4	num_syllables	1356	592	54	51	202.34	239.95
5	num_characters	6759	2996	164	199	1042.74	1130.41
6	num_stopwords	444	196	0	9	44.75	61.10
7	num_unique_words	254	442	8	42	136.84	169.37
8	flesch_reading_ease	117.26	116.45	-336.55	-382.23	105.15	53.02
9	flesch_kincaid_grade	172.50	190.00	-2.00	-1.70	2.64	22.80
10	avg_word_length	8.23	6.77	3.64	3.22	4.87	4.42
11	type_token_ratio	0.92	0.91	0.01	0.44	0.66	0.70
12	comma_count	23	57	0	0	0.14	0.73
13	period_count	222	97	2	0	13.33	7.72
14	exclamation_count	1	14	0	0	0.00	0.18
15	lexical_diversity	0.92	0.91	0.01	0.44	0.66	0.70

Table 8: Feature Statistics for AI and Human Texts for Arabic Essay (Subtask B)