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

Shifali Agrahari, Subhashi Jayant,
Saurabh Kumar, and Sanasam Ranbir Singh
Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
{a.shifali, j.subhashi, saurabh1003, ranbir}@iitg.ac.in

Abstract

Detecting AI-generated text in the field of academia is becoming very prominent. This paper presents a solution for Task 2: AI vs. Human – Academic Essay Authenticity Challenge in the COLING 2025 DAIGenC Workshop [1]. The rise of Large Language models (LLMs) like ChatGPT has posed significant challenges to academic integrity, particularly in detecting AI-generated essays. To address this, we propose a fusion model that combines pre-trained language model embeddings with stylometric and linguistic features. 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

The exponential growth of Large Language Models (LLMs) has led to widespread applications, including language translation, question answering, text generation, and beyond. However, their unauthorized use by students to complete homework, write essays, and write content-specific questions compromises academic integrity, highlighting the need for AI-driven LLM text detection. Using AI-generated content in academic contexts also poses challenges related to plagiarism (Liao, 2020).

The existing literature proposes various methods for AI-generated text detection, including feature-based models, supervised, zero-shot, and adversarial approaches. All of these models are designed to improve the result of detection in different languages and styles. Despite achieving decent overall accuracy, these methods still suffer from high false positives, where human-generated text is misclassified as AI-generated. Furthermore, class-wise accuracy remains a challenge, indicating room for improvement in distinguishing between human-generated text and AI-generated text.

To address these issues, The COLING 2025 Workshop on DAIGenC (Chowdhury et al., 2025) 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 task, framed as—*"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.

Our final model is a fusion of feature-based models and PLM embeddings. Initially, the PLM showed poor performance with a bias toward the majority class. By integrating linguistic and stylistic features, we improved the overall Macro F1 score. Our focus addressed three key challenges: capturing feature dependencies, handling class imbalance, and optimizing training to preserve linguistic representations in lower layers while enabling higher layers to capture task-specific (Essay) stylistic differences.

2 Related Work

Over the last few years, numerous approaches have been proposed to tackle the task of AI-generated text detection. Detecting machine-generated text is formulated primarily 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 general, 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.,

[1] https://gitlab.com/genai-content-detection/genai-content-detection-coling-2025

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 299–306
January 19, 2025.

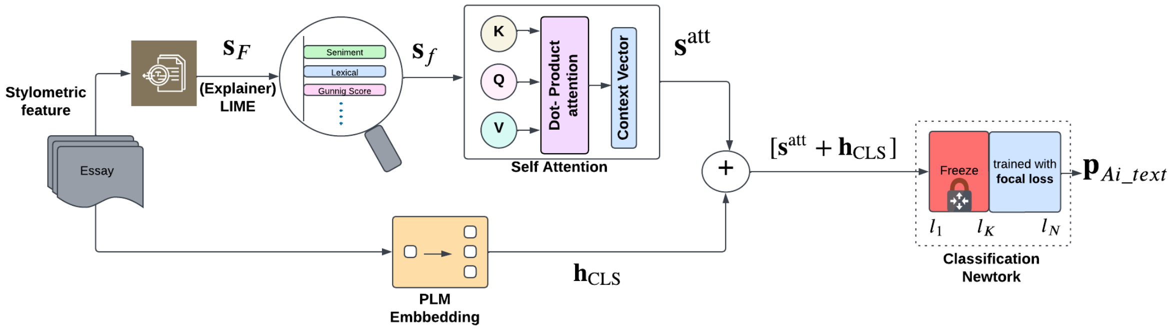

Figure 1: Proposed detector model architecture: fusion stylometric features with a PLM embedding.

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 the 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 Proposed Model

We use a fusion model that combines stylometric features with PLM embeddings, fine-tuned together for binary classification of AI vs. human text.

3.1 Stylometric Features

The stylometric features aim to capture different stylistic signals within a given text. As mentioned in Table 1, the stylometric features capture stylistic signals in three categories: Phraseology (how the author organizes words and phrases), Lexical Diversity (measures how varied the author's vocabulary), and Syntactic Diversity (author structured sentences and conveying emotions), definition of these features mentioned in Section A.2.1

3.2 Model

For each text input instance, we first extract the stylometric features as vector $\mathbf{s}_F \in \mathbb{R}^F$ where F is the number of stylometric features as mentioned in Table 1 then apply *LIME* (Local Interpretable model-agnostic Explanations) to select the most distinguishing feature as a vector $\mathbf{s}_f \in \mathbb{R}^f$, where f is the number of important features. These features help distinguish between human and AI-texts.

To capture the dependencies within the stylometric features, we apply a *self-attention* mechanism over the stylometric features, producing an attention-weighted vector $\mathbf{s}^{\text{att}} = \text{Attention}(\mathbf{s}_f)$. This attention function assigns weights to each stylometric feature based on its relevance to the dependency between the features.

In parallel, we obtain the CLS token embedding from the final hidden layer of the PLM, denoted as $\mathbf{h}_{\text{CLS}}$. This embedding captures the semantic meaning of the entire input text.

Next, we concatenate the attention-weighted stylometric vector $\mathbf{s}^{\text{att}}$ with the CLS token embedding $\mathbf{h}_{\text{CLS}}$ to create a combined feature vector $\mathbf{f}_{\text{concat}}$, defined equation 1. This vector is then passed through the classification network which is layer-wise freezing during fine-tuning. Let the PLM layers 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 $\theta_{l_1}, \ldots, \theta_{l_k}$ of the lower layer, which are initialized with pre-trained weights that preserve general linguistic representations, and update $\theta_{l_{k+1}}, \ldots, \theta_{l_n}$ for higher layers, as in equation 2, Here, k is a hyperparameter that determines how many of the lower layers of the pre-trained model remain frozen, retaining their general linguistic representations while the higher layers are fine-tuned.

$$\mathbf{f}_{\text{concat}} = [\mathbf{s}^{\text{att}}; \mathbf{h}_{\text{CLS}}] \quad (1)$$

$$L_{\text{fine-tune}} = \sum_{i=k+1}^{n} L(\theta_{l_i}) \quad (2)$$

The parameters $\theta_{l_{k+1}}, \ldots, \theta_{l_n}$ transform $\mathbf{f}_{\text{concat}}$

Stylometry Analysis	Feature Sets
Phraseology	Word count, Sentence count, Paragraph count, Mean, 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 (Mindner et al., 2023) (defined in A.2.1 and for detail result A.3)

into r, which is then passed through the final layer l_n for classification.

The final layer l_n generates the output representation $\mathbf{r}$, which is then passed through a softmax activation function to compute the class probabilities $p_\theta(y|\mathbf{r})$, where $y \in 0, 1$ indicates the class of the text (0 for "human-written" and 1 for "AI-generated"). The softmax function is defined as:

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

To address class imbalance, we apply focal loss, which modifies the 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:

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

Here, α is a balancing factor for class importance, and γ is a focusing parameter that down-weights easy examples. The focusing parameter 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 $(1 - p_\theta(y|\mathbf{r}))^\gamma$ reduces the loss for well-classified examples. The model is trained using focal loss and optimized through backpropagation.

In the testing phase, each text input instance is passed to the the trained model and the output $\mathbf{r}$ is processed by the softmax function to predict the class $\hat{y} = \arg\max_y p_\theta(y|\mathbf{r})$.

Model performance is evaluated using accuracy, Macro precision, Macro recall, and Macro F1-score which are discussed in Results section.

4 Experiments

4.1 Dataset

For each task, there are three datasets provided by (Chowdhury et al., 2025): Train, Dev and Test. Training and development data with labels (AI or human) for the development phase and for the evaluation phase, testing data without labels for both tasks. All descriptions with respect to the size of data set is mentioned in Table 2.

Data	Train		Dev		w/o label	
	#AI	#Human	#AI	#Human	#Dev	#Test
English	1467	629	391	1235	567	1130
Arabic	925	1145	299	182	293	886

Table 2: Dataset count distribution across training, development, and testing set.

4.2 Experimental Setup

For both Subtasks, the hyperparameters include an epoch size ranging from 50 to 250, while the batch size is fixed at 32, determined by the available GPU resources. Further details of the experimental setup are presented in Section A.1.

4.3 Feature and Model Selection

To improve model interpretability, we use **LIME** as mentioned (Ribeiro et al., 2016) for feature selection, helping identify the most influential features for detecting AI-generated text. Feature details of LIME are presented in Appendix A.2.1.

For subtask A (English essays), calculate the linguistic and stylometric characteristics mentioned in Table 1. LIME highlights such as average sentence length, number of stop words, type token ration,

model	Feature	F1
Baseline (n-gram)	-	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 of Macro F1 on test Data with and without Features for Subtask A English

etc., are the top 12 most discriminative characteristics. For subtask B (Arabic essays), 11 features such as Sentiments and Flesch reading ease are highly discriminative features after applying LIME. However, certain features, such as part-of-speech (POS) tags are less straightforward in Arabic due to its rich morphology, lack of strict word order, and complex inflectional system compared to languages like English.. Details of features are given in Section A.3.

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 understanding the Arabic language.

5 Results and Analysis

Table 3 (for English) and Table 4 (for Arabic) show the results of the test dataset. The baseline results were provided by the organizer, while all other results are based on our experimental findings. For Subtask A, our proposed model, the fusion of *DeBERTa-base* and the symmetry characteristics, achieves the highest score of 0.978 on testing dataset. For Subtask B, our proposed model, Fusion of *AraBERT* and Stylometry features, achieves the best performance with an F1 score of 0.9429. Notably, in Subtask A, other models also show competitive performance when combined with features. In Subtask B, *AraBERT* without features achieves an impressive F1 score of 0.9214, leveraging its design tailored to the Arabic language to effectively capture its unique linguistic features. Such Arabic-specific models are optimized for the language's morphology and syntax, often providing slight performance advantages in specialized tasks. Figure 2 illustrates the confusion matrix for the development dataset using our proposed models for both subtasks. It can be observed that Arabic data tend to be misclassified more frequently compared to English data.

Table 5 highlights the strong performance of our final models, which secured 10th position in Subtask A (English) and 13th position in Subtask B (Arabic) in the official task rankings.

model	Feature	F1
Baseline (n-gram)	-	0.4605
XLM-RoBERTa-base	-	0.9188
AraBERT v02	-	0.9214
XLM-RoBERTa-base	Yes	0.9414
AraBERT v02	**Yes**	**0.9429**

Table 4: model performance of Macro F1 on Test Data with and without Features for Subtask B Arabic

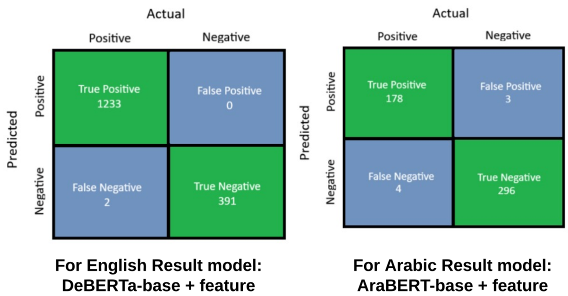

Figure 2: Performance Metrics on development Dataset

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: Leadboard Score of Our Final model

6 Conclusion

The unethical misuse of LLMs in academic contexts poses challenges to integrity, highlighting the need for effective AI-generated text detection. Our fusion model, combining stylometric features with PLM embeddings, addresses 3 key challenges identifying highly discriminative ones using LIME, focal loss for addressing class imbalance and apply layer-wise freezing during fine tuning to capture task-specific stylistic differences in essays. These strategies have significantly improved model performance. For Subtask A (English), our DeBERTa + features model achieved a Macro F1 score of 0.978, while for Subtask B (Arabic), the AraBERT + features model scored 0.9429. Future work may refine these techniques to further enhance model's classwise F1 and generalization.

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.

Shammur Absar Chowdhury, Hind Al-Merekhi, Mucahid Kutlu, Kaan Efe Keleş, Fatema Ahmad, Tasnim Mohiuddin, Georgios Mikros, and Firoj Alam. 2025. GenAI content detection task 2: AI vs. human – academic essay authenticity challenge. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, Abu Dhabi, UAE. International Conference on Computational Linguistics.

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of* the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

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

As mention in Table 6, We employ two experimental setups. In the first, we fine-tune the Pre-Trained Language model (PLM) independently for each subtask over 50 epochs, using the Adam optimizer with a learning rate of 2×10^{-5} and L2 regularization (weight decay 0.01). The second setup uses the PLM for training with batch normalization, and 0.5 dropout. The model is trained with a 2×10^{-5} learning rate, L2 regularization of 0.01, and early stopping after 25 epochs. Focal loss addresses class imbalance, emphasizing hard-to-classify examples. All experiments are implemented in PyTorch (Paszke et al., 2019), for efficient training and handling of large datasets.

Hyperparameter	**Setup: Fine-tuning PLM**
Epochs	10-250
Batch Size	5
k	6 layer
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.

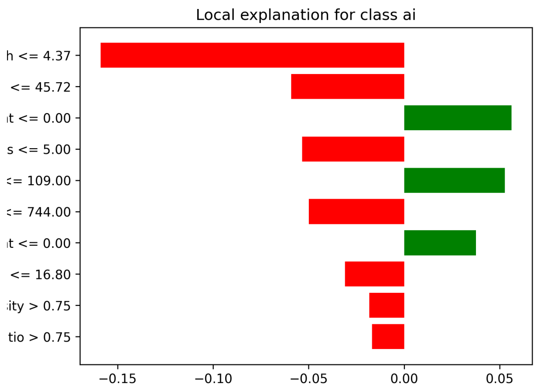

Figure 3: LIME Explanation for Subtask B as mentioned in (Ribeiro et al., 2016)

A.2 Stylometry Analysis Feature Sets

A.2.1 Phraseology

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 frequently punctuation marks are used.

A.2.2 Lexical Diversity

- **Type-Token Ratio (TTR):** A measure of lexical variety, ratio of UWC and WC, where UWC is the number of unique words and WC is the total word count.

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

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

- **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.

A.2.3 Syntactic Diversity

Sentiment Polarity measure of the emotional tone of the text, ranging from -1 (-ve) to 1 (+ve).Sentiment Subjectivity measure of how subjective or opinion-based the text is, usually ranging from 0 (objective) to 1 (subjective).

A.3 Features Analysis of English & Arabic

Table 8 and Table 7 compare linguistic and stylometric features between AI-generated and human-written essays in English and Arabic. For instance, in English essays, AI texts exhibit higher average word counts (321.37 vs. 254.0) and sentence counts (13.22 vs. 9.0). Similarly, in Arabic essays, AI texts display longer average word counts (215.11 vs. 251.17) but fewer unique words (136.84 vs. 169.37). Other features, such as readability scores (e.g., Flesch Reading Ease), sentiment metrics, and part-of-speech proportions, indicate stylistic differences, highlighting AI's more mechanical and less nuanced language use compared to humans.

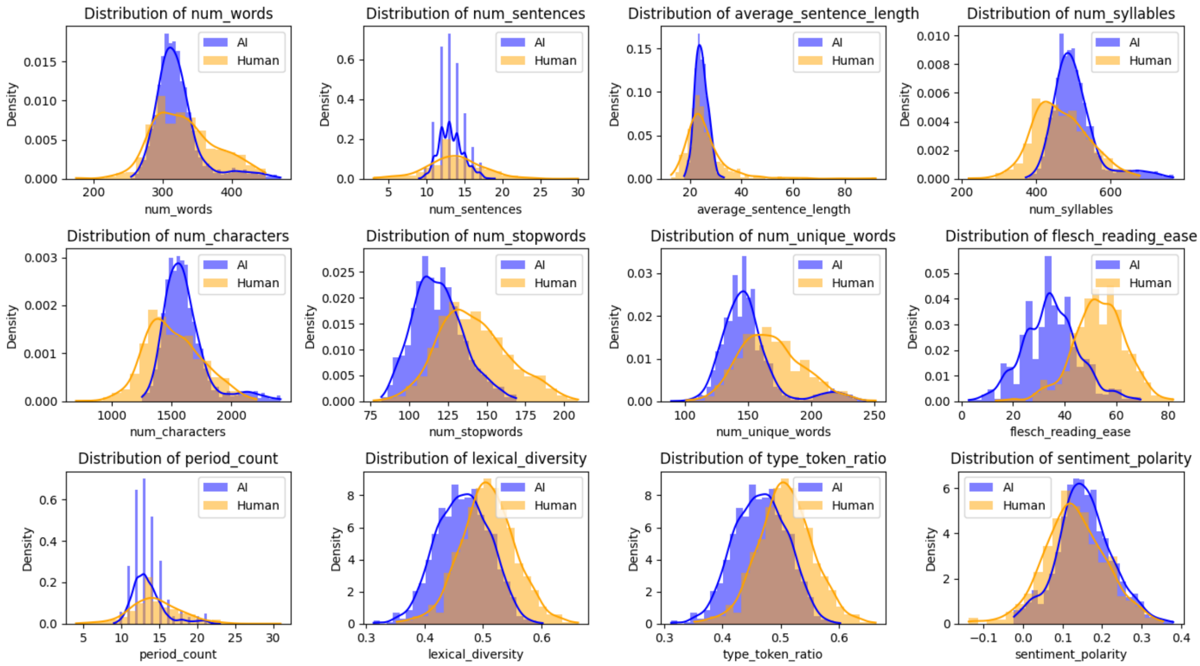

Figure 4: Distribution of features for AI and Human labels.

#	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	avg_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 7: Feature Statistics for AI and Human Texts for Arabic Essay (Subtask B).

#	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 8: Linguistic and Stylometric Features Comparison in English Essays.