# Team EssayDetect at GenAI Detection Task 1: Multilingual MGT Detection: Leveraging Cross-Lingual Adaptation for Robust LLMs Text Identification

**Anonymous ACL submission**

## Abstract

Detecting AI-generated text has become increasingly prominent. This paper presents our solution for the GenAI Content Detection Task 1 Subtask 2, where we address the challenge of distinguishing human-authored text from machine-generated content, especially in multilingual contexts. We introduce MLDet, a model that leverages Cross-Lingual Adaptation and Model Generalization strategies for Multilingual Machine-Generated Text (MGT) detection. By combining language-specific embeddings with fusion techniques, MLDet creates a unified, language-agnostic feature representation, enhancing its ability to generalize across diverse languages and models. Our approach demonstrates strong performance, achieving macro and micro F1 scores of 0.7067 and 0.7187, respectively, and ranking 15th in the competition. We also evaluate our model across datasets generated by different distinct models in many languages, showcasing its robustness in multilingual and cross-model scenarios.

## 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 1: Binary Multilingual Machine-Generated Text Detection (Human vs. Machine)"* aim to refresh training and testing data with generations from novel LLMs and include new languages. The task, framed as follows—*"determining whether a given text is generated by a machine or authored by a human"*—is a binary classification challenge divided into two sub-tasks: Subtask B: Multilingual MGT detection.

We evaluated multiple approaches and ultimately focused on one strategies. our approach for multilingual Machine-Generated Text (MGT) detection, which includes Cross-Lingual Adaptation and Model Generalization strategies. This methodology leverages language-specific embeddings, cross-lingual fusion, and model-invariant features to improve generalization across languages and models. Our model, **MLDet**, demonstrates a balanced performance on both macro and micro F1 scores, achieving 0.7067 and 0.7187, respectively and 15th rank in The GenAI Content Detection Task 1 Subtask 2

## 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 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., 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

In this section, we outline our approach for multilingual Machine-Generated Text (MGT) detection, which includes Cross-Lingual Adaptation and Model Generalization strategies. This methodology leverages language-specific embeddings, cross-lingual fusion, and model-invariant features to improve generalization across languages and models.

### 3.1 Dataset Description

The dataset consists of samples across multiple languages and models, organized with the following columns:

- **source**: Specifies the source of the text (e.g., human-generated or machine-generated).
- **sub_source**: Specifies any additional categorization or source subtype.
- **lang**: Indicates the language of the text, represented by language codes (e.g., EN for English, ZH for Chinese).
- **model**: The model used to generate the machine-generated text.
- **label**: Binary label indicating whether the text is human-generated (0) or machine-generated (1).
- **text**: The textual content itself.

### 3.2 Language-Specific Embedding Extraction

For each language $l$ in the dataset, we utilize a pre-trained embedding model $M_l$ specialized for that language, such as RoBERTa for English, Chinese-BERT for Chinese, and AraBERT for Arabic, as mentioned in Table 6. Given an input text $x^{(l)}$ in language $l$, we obtain a feature vector $h^{(l)}$ using the corresponding embedding model as defined Eq.1

$$h^{(l)} = M_l(x^{(l)}) \tag{1}$$

This produces a feature vector that captures both language-specific and general semantic features.

### 3.3 Cross-Lingual Fusion for Unified Representation

To create a language-agnostic representation, we fuse embeddings across languages. Let $H = \{h^{(l_1)}, h^{(l_2)}, \ldots, h^{(l_n)}\}$ represent feature embeddings across languages.

- **Concatenation Fusion**: Concatenate embeddings from different languages:

$$h_{\text{fusion}} = [h^{(l_1)}; h^{(l_2)}; \ldots; h^{(l_n)}] \in \mathbb{R}^{n \times d} \tag{2}$$

- **Weighted Summation Fusion**: Alternatively, we apply a weighted summation where each language embedding $h^{(l)}$ is scaled by a learnable weight $w^{(l)}$:

$$h_{\text{fusion}} = \sum_{l \in \mathcal{L}} w^{(l)} h^{(l)} \tag{3}$$

### 3.4 Cross-Lingual Consistency Loss

To enforce consistency across languages, we introduce a cross-lingual consistency loss that encourages similarity between embeddings of the same sample across languages. For each pair of languages $(l_i, l_j)$:

$$L_{\text{cross-lingual}} = \frac{1}{|\mathcal{L}|(|\mathcal{L}|-1)} \sum_{i \neq j} \|h^{(l_i)} - h^{(l_j)}\|^2$$

This loss aligns embeddings across languages, promoting language-invariant features.

### 3.5 Model Generalization for Machine-Generated Text Detection

Given that the training set includes 43 models and the testing set includes 20 different models, we introduce a model generalization loss to reduce reliance on specific training models.

- For each model $M_m$, obtain a feature vector $h_m$ for a text $x$:

$$h_m = M_m(x)$$

- **Cross-Model Pairwise Loss**: To promote model-invariant features, minimize divergence between embeddings from different models:

$$L_{\text{model-gen}} = \frac{1}{|\mathcal{M}|(|\mathcal{M}|-1)} \sum_{m \neq m'} \|h_m - h_{m'}\|^2$$

- **Noise Augmentation**: During training, add Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to simulate unseen models:

$$h_m^{\text{aug}} = h_m + \epsilon$$

### 3.6 Total Loss Function

Our model is optimized with a combination of label classification, cross-lingual, and model generalization losses. The total loss function is given by:

$$L_{\text{total}} = \alpha L_{\text{label}} + \beta L_{\text{cross-lingual}} + \gamma L_{\text{model-gen}}$$

where $L_{\text{label}}$ is the binary cross-entropy loss for classification, and $\alpha$, $\beta$, and $\gamma$ are hyperparameters controlling the contributions of each component.

### 3.7 Training and Evaluation

We train the model by minimizing $L_{\text{total}}$ with gradient descent. During evaluation, we use the Macro F1 score to assess balanced performance across classes (human-generated vs. machine-generated text), validating the model's generalization capability across unseen languages and models.

## 4 Experiments

### 4.1 Data

Shared task organizers have been used for Subtask B: Multilingual MGT detection. There are three datasets: Training and development data with 7 columns id, source, sub_source, lang model label and text; for the development phase, they provided Testing data for the Evaluation phase. All descriptions mentioned in Table 1

The dataset exhibits differences in the number of unique models, languages, and domains, as summarized in Table 7. Specifically, the text in the training and development datasets is generated using 43 distinct models (Section A.3), while the training dataset uses 20 different models (Table 9). Additionally, the training dataset includes data in 9 languages (Table 4), whereas the testing dataset contains text in 20 languages (Table 8). These variations in models and languages are essential for training and evaluation processes.

| Data | AI | Human | Total |
|---|---|---|---|
| Train | 674083 | 257968 | 674083 |
| Dev | 178728 | 110166 | 288894 |
| Test | 77791 | 73634 | 151425 |

Table 1: Data for AI, Human, and Total across Train, Dev, and Test datasets.

### 4.2 Experimental setup

Table **??** 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 all calculate accuracy and F1 score Further details of experimental setup in presented in section A.1.

### 4.3 Model Training

In this experiment, we use a multilingual dataset, as outlined in Section 3.2. For each language, a specific embedding model is used: for example, *Chinese-BERT* (Sun et al., 2021) for Chinese and *AraBERT* (Antoun et al., 2020) for Arabic, as detailed in Table 6. For training, we consider pretrained language models such as *XLM-RoBERTa* (Wiciaputra et al., 2021) and *mBERT* (Wu and Dredze, 2020). As shown in Table 2, different types of model training are applied. The first two models involve direct fine-tuning of pretrained language models (PLMs). The *mBERT + CM* model is trained with cross-model adaptation, as described in Section 3.5, while the *mBERT + CL* model is trained using Cross-Lingual Fusion in Section **??**. Finally, the ***MLDet*** model incorporates Cross-Lingual Adaptation and Model Generalization strategies, as described in Section 3.

## 5 Results

The results presented in Tables 2 and 3 provide a comprehensive analysis of the performance of various models on multilingual Machine-Generated Text (MGT) detection, with a focus on micro F1, accuracy, and macro F1 scores. On the Dev dataset,

*XLM-RoBERTa* scores 0.4631 for micro F1 and accuracy, while *mBERT* improves to 0.8352. The *mBERT + CM* model reaches 0.8523, demonstrating the benefit of Cross-Model Adaptation. The *mBERT + M* model scores slightly lower. Our custom model, **MTDet**, achieves 0.7938, outperforming *XLM-RoBERTa*, but not *mBERT + CM*.

| Model | Score | Micro F1 | Accu. |
|---|---|---|---|
| **XLM-RoBERTa** | 0.4133 | 0.4631 | 0.4631 |
| **mBERT** | 0.5203 | 0.8352 | 0.8352 |
| **mBERT + cm** | 0.5832 | 0.8523 | 0.8521 |
| **mBERT + M** | 0.6044 | 0.8264 | 0.8264 |
| **MTDet (Ours)** | **0.7739** | **0.7938** | **0.7938** |

Table 2: Performance scores for different models on the Dev Dataset.

When evaluating on the test dataset, *XLM-RoBERTa* again shows relatively low performance, with a macro F1 score of 0.3876 and micro F1 score of 0.6798. In contrast, both *mBERT + CM* (Cross-Model Adaptation) and *mBERT + CL* (Cross-Lingual Fusion) exhibit strong performance, with *mBERT + CL* achieving the highest micro F1 score of 0.8650. Our model, **MLDet**, demonstrates a balanced performance on both macro and micro F1 scores, achieving 0.7067 and 0.7187, respectively. Although it does not reach the highest micro F1 score, its macro F1 performance suggests a more balanced generalization across different languages and domains, reflecting its robustness in multilingual MGT detection.

| Test Model | Macro F1 | Micro F1 |
|---|---|---|
| **XLM-RoBERTa** | 0.3876 | 0.6798 |
| **mBERT** | 0.4307 | 0.7135 |
| **mBERT + CM** | 0.5678 | 0.8123 |
| **mBERT + CL** | 0.4897 | 0.8650 |
| **MLDet (Ours)** | **0.7067** | **0.7187** |

Table 3: Performance comparison of various test models on Macro and Micro F1 scores.

## 6 Analysis

The performance of our **MLDet** model, as seen in Tables 2 and 3, demonstrates the effectiveness of integrating advanced cross-lingual adaptation and model generalization strategies. Our model's design incorporates multiple adaptation techniques tailored to multilingual Machine-Generated Text (MGT) detection, which has contributed to a more balanced and generalized performance across diverse languages and domains.

As illustrated in Table 7, the training and development (Dev) sets each include nine unique languages, 43 models, and 36 domains. This diversity enables the model to learn from varied linguistic structures and contextual nuances across languages. For the test set, the number of unique languages increases to 16, with 20 models and 27 domains, presenting a more challenging scenario. Our approach to cross-lingual adaptation addresses these challenges by enabling the model to generalize its understanding beyond the training languages, capturing unique patterns in unseen languages and domains.

A significant advantage of **MLDet** lies in its combined use of *Cross-Lingual Adaptation* and *Model Generalization* techniques. Cross-lingual adaptation facilitates the transfer of learned knowledge between languages, allowing the model to adapt effectively even when faced with unseen language pairs. Moreover, the model generalization strategies enhance the ability of **MLDet** to perform consistently well on new domains, as evidenced by its balanced macro F1 and micro F1 scores. Unlike traditional fine-tuning, which may be limited by language-specific biases, these generalization techniques prevent overfitting to specific language features, thereby promoting adaptability across both seen and unseen languages.

## 7 Conclusions

In conclusion, the robust performance of **MLDet** on diverse multilingual datasets underscores the importance of incorporating cross-lingual adaptation and model generalization strategies. By capturing a wide range of linguistic and contextual information, these strategies allow the model to generalize effectively across languages and domains, positioning **MLDet** as a versatile and efficient solution for MGT detection in multilingual settings.

## 8 Limitations

While MLDet demonstrates strong performance, several limitations remain. First, its effectiveness is constrained by biases in the training datasets, which may not fully capture all linguistic or stylistic variations. The reliance on language-specific embeddings also limits its application to low-resource languages or dialects. Additionally, the model's

generalization ability is challenged by novel content generation techniques. Finally, the computational demands of cross-lingual fusion and model generalization hinder scalability for real-time applications. Future work should focus on addressing these challenges through enhanced datasets, fine-tuning for low-resource languages, and optimizing for faster inference.

## 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.

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*.

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*.

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.

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*.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. *arXiv preprint arXiv:2106.16038*.

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).

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? *arXiv preprint arXiv:2005.09093*.

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

The experimental setup for this study includes a comprehensive range of hyperparameters, multilingual datasets, and model embeddings tailored to effectively detect machine-generated text across diverse languages and domains.

| Language (Code) | Human | AI |
|---|---|---|
| *Arabic (ar)* | 344 | 1770 |
| *Bulgarian (bg)* | 4205 | 3886 |
| *German (de)* | 231 | 4462 |
| *English (en)* | 229209 | 381467 |
| *Indonesian (id)* | 1895 | 2081 |
| *Italian (it)* | 0 | 4174 |
| *Russian (ru)* | 684 | 630 |
| *Urdu (ur)* | 2085 | 1676 |
| *Chinese (zh)* | 19315 | 15969 |

Table 4: Counts of Human and AI instances across languages in Training Dataset.

Key hyperparameters, such as learning rate, batch size, and dropout rate, were carefully tuned to optimize model performance. Additionally, weights for classification, domain, and cross-lingual loss were experimentally adjusted to ensure the model's adaptability to varied linguistic structures. The optimizer used was Adam, with specific parameters for Beta values and epsilon, while learning rate scheduling was customized based on experimental results. The setup is designed to capture fine-grained cross-lingual features, thereby enabling robust language-specific and language-agnostic pattern recognition.

| Hyperparameter | Typical Values |
|---|---|
| Learning Rate ($\eta$) | $1e-5$ to $1e-3$ |
| Batch Size | 16, 32, 64 |
| Number of Epochs | 100 to 500 |
| Weight for Classification Loss ($\lambda$) | Tuned based on experiment |
| Weight for Domain Loss ($\gamma$) | Tuned based on experiment |
| Weight for Cross-Lingual Loss ($\delta$) | Tuned based on experiment |
| Dropout Rate | 0.1 to 0.5 |
| Gradient Reversal Layer Parameter | Tuned based on experiment |
| Hidden Layer Dimensions | Tuned based on experiment |
| Optimizer (Adam Parameters) | Beta1: 0.9, Beta2: 0.999, Epsilon: $1e-8$ |
| Learning Rate Scheduler Parameters | Tuned based on experiment |

Table 5: List of Hyper parameters for the Experiment during Training

| Language | Embedding |
|---|---|
| English | *RoBERTa* |
| Chinese | *Chinese-BERT* |
| Bulgarian | *XLM-RoBERTa* |
| German | *GottBERT* |
| Italian | *AlBERTo* |
| Indonesian | *IndoBERT* |
| Urdu | *UrduBERT* |
| Arabic | *AraBERT* |
| Russian | *RuBERT* |

Table 6: Languages and their corresponding embeddings during Training.

The dataset is meticulously organized by language, embedding models, and instance counts for both human and AI-generated content, reflect-

| | lang | model | domain |
|---|---|---|---|
| Train | 9 | 43 | 36 |
| Dev | 9 | 43 | 36 |
| Test | 16 | 20 | 27 |

Table 7: Table showing the different type of unique lang, model, and domain.

ing a highly diverse and balanced approach. The training dataset spans nine languages, each paired with a language-specific embedding model (e.g., RoBERTa for English, Chinese-BERT for Chinese, and AraBERT for Arabic). This arrangement enables nuanced feature extraction tailored to each language, enhancing the model's ability to distinguish between human and AI text in a multilingual context. The distribution of human and AI instances across languages highlights a well-rounded dataset, with larger counts in commonly studied languages like English and Chinese and sufficient representation in languages like Bulgarian, Indonesian, and Urdu.

| Language | Human | AI |
|---|---|---|
| Arabic | 4350 | 6320 |
| Chinese | 29947 | 33062 |
| Dutch | 600 | 600 |
| German | 1865 | 0 |
| Hebrew | 1182 | 0 |
| Hindi | 599 | 600 |
| Indonesian | 600 | 600 |
| Italian | 2496 | 2800 |
| Japanese | 300 | 300 |
| Kazakh | 1171 | 1300 |
| Norwegian | 1544 | 0 |
| Russian | 13039 | 13094 |
| Spanish | 600 | 600 |
| Urdu | 13190 | 17315 |
| Vietnamese | 1126 | 1200 |
| Russian (2) | 1025 | 0 |

Table 8: Counts of Human and AI instances across various languages in Testing Dataset.

### A.2 Testing Dataset details

The testing dataset expands the linguistic range, incorporating additional languages such as Kazakh, Norwegian, and Hindi, thus testing the model's capacity to generalize to unseen linguistic contexts. A broad array of generative models, including recent releases like GPT-4, Llama, and Baichuan, are represented, allowing a thorough evaluation of the model's effectiveness across diverse AI text generation systems. This experimental design facilitates a detailed assessment of the model's cross-lingual performance and robustness against various language models, ensuring comprehensive validation of the approach.

| Model | Count |
|---|---|
| Human | 73634 |
| GPT-4o | 28538 |
| GPT-4o-mini | 6845 |
| gpt4o | 6591 |
| Vikhrmodels | 6503 |
| gpt-4o-2024-05-13 | 5998 |
| Baichuan2-13B-Chat | 5521 |
| ChatGLM3-6B | 5359 |
| Llama 3.1 405B instruct | 4000 |
| gpt-4o | 2400 |
| gpt-4 | 1545 |
| GPT-4-turbo | 1400 |
| glm-4-9b-chat | 778 |
| claude-3-5-sonnet | 773 |
| GPT4 | 299 |
| Qwen | 297 |
| GPT3.5 | 297 |
| ChatGLM | 295 |
| Baichuan | 283 |
| qwen2.5 72b | 69 |

Table 9: Counts of instances for different models in Testing Dataset.

### A.3 Details of AI Text Generation Models

The experimental setup includes a variety of language generation models, each with unique capabilities. The "Human" data serves as a baseline for evaluating AI-generated text. Models like `GPT-3.5-Turbo` and `GPT-35`, developed by OpenAI, are known for their advanced text generation capabilities and context awareness. `Davinci` and its variants (`text-davinci-003`, `text-davinci-002`) are also part of OpenAI's GPT-3 series, excelling in handling complex instructions. `Bloomz` and its variant `bloom_7b` are part of the BLOOM model series, which is multilingual and open-source, supporting many languages. `Cohere` focuses on commercial applications, while `Mixtral-8x7b` and `Gemma-7b-it` are specialized for specific languages and regions. The `Llama3` models, including `Llama3-8b` and `Llama3-70b`, are Meta's efficient transformers, optimized for

diverse tasks. GPT4 and GPT4o, the latest in OpenAI's GPT series, exhibit enhanced reasoning capabilities. Dolly and Dolly-v2-12b, developed by Databricks, offer customizable open-source models, and Gemma2-9b-it is specifically tuned for Italian. The T0 series models, such as T0_3b and T0_11b, are designed for zero-shot learning across domains. The Flan_T5 models, including small to extra-large variants, are fine-tuned T5 models that excel in few-shot tasks. Meta's OPT series, ranging from 1.3b to 30b parameters, offers open pre-trained models, while the 13B, 30B, 65B, and 7B models are large-scale transformer variants. GLM130B, a 130 billion parameter model, offers high performance at scale. GPT-NeoX and GPT-J are open-source models, designed by EleutherAI, for robust text generation. Finally, Llama2-fine-tuned is a fine-tuned version of the Llama series, and Jais-30b is a model focused on Arabic language tasks.