MuST: Multilingual Span-Level Machine-Generated Text Detection

Anonymous authors
Paper under double-blind review

Abstract

Detecting machine-generated text (MGT) is critical, as the misuse of large language models (LLMs) has raised concerns about the integrity, authenticity, and trustworthiness of information. In real-world scenarios, the task is challenging because human–AI collaboration produces hybrid texts, motivating the development of fine-grained detection of MGT. This line of research performs well on the English corpus, but its generalization to other languages is under-exploited. To address this challenge, we introduce MuST, a framework for multilingual fine-grained MGT detection. Our framework features unified modeling for two fine-grained MGT tasks, i.e., span-level detection and boundary detection. Besides, to accommodate different languages, we design augmented word embeddings by combining the information of neighboring words, N-grams, the sentence, and the normalized position. To assess our approach under multilingual settings, we further propose MuST-Mix, a fine-grained dataset for multilingual MGT detection. Evaluation shows that MuST outperforms 11 SOTA detectors in eight languages and is robust to unseen generators and unseen domains.

1 Introduction

The abuse of large language models (LLMs) may undermine the authenticity, reliability, and trustworthiness in many domains (Wu et al., 2025; Ghosal et al., 2023; Yang et al., 2024). For example, LLMs can be used to generate misinformation and fake news on a large scale (Pan et al., 2023), produce deceptive content for scams and fraud (Heiding et al., 2024), and contaminate the online information ecosystem with large volumes of synthetic content (Hanley & Durumeric, 2024). Hence, machine-generated text (MGT) detection is increasingly significant for trust, accountability, and domain-specific standards (Wang et al., 2024a; Dugan et al., 2024).

The task, however, is challenging in real-world scenarios. First, humans often collaborate with LLMs in text generation (Mysore et al., 2025). The hybrid text, therefore, illustrates both human and AI characteristics. As a result, fine-grained detection techniques identifying the provenance of each text span are required to distinguish human and LLM contributions (Su et al., 2025; Jiang et al., 2025). Second, due to the multilingual nature of LLMs, humans may use LLMs to generate text in their language. Hence, the generalization across multiple languages is essential to guarantee a wider application (Macko et al., 2023; Wu et al., 2026).

Existing techniques for fine-grained MGT detection can be categorized into span-level detection (Wang et al., 2023; Jiang et al., 2025; Su et al., 2025) and boundary detection (or boundary localization in some literature) (Kushnareva et al., 2024; Zeng et al., 2024; Zhang et al., 2024b; Tolstykh et al., 2026a; Lei et al., 2025; Sai Teja et al., 2026). Though inspiring, these approaches are designed and evaluated in English settings. And their generalization across different languages remains unknown. On the other hand, a number of detection methods have been developed for multilingual settings (Ben-fares et al., 2024; Spiegel & Macko, 2024; Xiong et al., 2024; Ta et al., 2026). These work, however, formulates the detection task at the document level as binary or multiclass classification. They are not designed for span-level detection.

This paper presents MuST, a unified framework for fine-grained MGT detection. The core idea of our approach is twofold. First, boundary detection (identifying the boundary locations of MGT and human-written text, HWT) and span-level detection (attributing the authorship of each side) are complementary to each other. And both are critical to analyzing hybrid texts. We therefore propose

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a unified loss function that jointly optimizes the two tasks. Second, individual words contain little information for distinguishing MGT and HWT. Hence, language-specific contextual information is necessary to improve the detection performance. To cover a broad range of languages, we therefore enrich the representation of a focal word by combining the embeddings of its adjacent words, N-grams, the sentence, and its normalized position. The evaluation shows that MuST successfully identifies the boundary of MGT and HWT and accurately estimates the span-level attribution across eight languages and unseen generators and domains. Compared to the SOTA detector, MuST improves boundary type F1 score by **+18.50%** (0.7269→0.8614), and position F1 by **+34.50%** (0.5928→0.7973).

Our main contributions are thus as follows: (1) We propose MuST, a framework for fine-grained MGT detection by jointly optimizing span-level detection and boundary detection; (2) Based on existing multilingual human- and AI-generated content, we propose MuST-Mix, a span-level human-AI hybrid benchmark for multilingual fine-grained MGT detection; (3) Evaluation on eight languages shows that MuST outperforms 11 SOTA detectors in fine-grained MGT detection tasks, and it is robust to unseen generators and domains.

2 Related Work

Prior work has approached fine-grained MGT detection from two perspectives: document-level fine-grained labeling and span-level detection. First, at the document level, some studies estimate the extent of LLM involvement (Liang et al., 2025), whereas others frame detection as a multiclass classification task (Zhang et al., 2024a; Abassy et al., 2024; Bao et al., 2025; Ta et al., 2026; He et al., 2025). These approaches, however, limit the detection to an entire document or corpus and can hardly be adapted to short texts. Second, span-level approaches identify authorship within a document. They either address span-level detection (Su et al., 2025; Wang et al., 2023; Jiang et al., 2025), or boundary detection (Tolstykh et al., 2026a; Zeng et al., 2024). These approaches perform well on the English corpus, while their generalization across languages remains unknown (Tolstykh et al., 2026a). Our paper aligns with the second perspective. Different from prior work, our framework jointly addresses both detection tasks and focuses on multilingual settings, which is more challenging. Multilingual detection methods generalize towards multiple languages through fine-tuned models (Spiegel & Macko, 2024; Ta et al., 2026) and adversarial training (Li et al., 2026). However, these methods are designed for document-level classification only.

Benchmarks for MGT detection can be categorized into two types: (1) document-level classification tasks and (2) span-level detection tasks. Document-level datasets usually provide human–machine paired samples with binary (Guo et al., 2023; Bevendorff et al., 2025; Qing et al., 2026) or multiclass labels (Zhang et al., 2024a; Bao et al., 2025; Artemova et al., 2025; Abassy et al., 2024; Saha & Feizi, 2025; Ta et al., 2026; Mysore et al., 2025). Span-level benchmarks, on the other hand, provide annotations for each text span, indicating whether it is human- or machine-generated (Dugan et al., 2023; Kushnareva et al., 2024; Su et al., 2025). The aforementioned datasets, however, provide monolingual samples only. Multilingual datasets provide annotations at the document level (Wang et al., 2024b; Macko et al., 2023; 2025; Wu et al., 2026; Ta et al., 2026), leaving the span-level detection task under-exploited. The most similar research to ours is LLMTrace (Tolstykh et al., 2026b), which provides multilingual span-level annotations. Differently, LLMTrace focuses on English and Russian, while MuST-Mix is composed of span-level samples from eight languages, including English, Chinese, German, French, Spanish, Russian, Arabic, and Portuguese.

To distinguish MGT and HWT at the span level, a well-designed word embedding is necessary. Word embeddings can be augmented with adjacent words (Devlin et al., 2019; Raffel et al., 2020; Ma et al., 2023), N-grams (Diao et al., 2020; Song et al., 2021; Qu et al., 2026), sentences (Patel & Domeniconi, 2024), and positions (Mensah et al., 2021; Yang et al., 2025). Inspired by these approaches, MuST builds word embeddings by combining the information from adjacent words, n-grams, sentences, and the position. And it uses an MLP to combine the complementary information with the Contextual Encoder.

3 MuST-Mix Construction

We build MuST-Mix based on five document-level HWT–MGT paired datasets, including DetectRL-X (Wu et al., 2026), HC3 (Guo et al., 2023), LLMTrace (Tolstykh et al., 2026b), M4GT

Task 3 (Wang et al., 2024a), and C-ReD (Qing et al., 2026). These datasets cover multilingual text, multiple generators, and heterogeneous domains, providing diverse annotations for MuST-Mix.

The core idea of dataset construction is semantic matching and controlled replacement. *Semantic Matching:* After data cleaning and preprocessing, we obtain MGT—HWT document pairs in which the two documents contain the same number of paragraphs, and their paragraphs are aligned one-to-one. For each pair of paragraphs, we perform semantic matching at the sentence level. We first encode every contiguous group of one to three sentences in the two paragraphs with Qwen3-Embedding (Zhang et al., 2025). We write $H_{p:q}$ for the block of HWT sentences from $p+1$ to q, and $M_{p:q}$ for the same notation applied to the MGT rewrite, where $E(\cdot)$ is the group encoder and $\cos(\cdot,\cdot)$ is the cosine similarity between two group embeddings. Then, based on HWT, we compute the cost of replacing the first i sentences in HWT with the first j sentences in MGT, and use dynamic programming to find the optimal replacement cost. The cost is formally defined as,

$$Cost_{i,j} = \min_{h,a}\Big[Cost_{i-h,j-a} + 1 - \cos(E(H_{i-h:i}), E(M_{j-a:j})) + \lambda_{\text{ov}}(h+a-2)\Big], \quad (1)$$

where $1 \le h \le \min(3,i)$ and $1 \le a \le \min(3,j)$. The term $1-\cos(E(H_{i-h:i}), E(M_{j-a:j}))$ means that a pair of blocks with higher embedding similarity receives a smaller penalty. The last term, $\lambda_{\text{ov}}(h+a-2)$, grows with the number of overlapping sentences in a matched block beyond the one-to-one case.

Controlled Replacement: To diversify the number of human–AI transitions, we control the number of replacement sentences in each HWT paragraph. Specifically, we replace one, two, or three sentences in HWT with their semantically matched sentences in MGT. The number of human-AI transitions ranges from zero to six in the resulting paragraphs. After the replacement, we then assign a label to every word and every sentence-level boundary. The detailed algorithm is given in Appendix A.3. Finally, we obtain MuST-Mix, of which the statistics are shown in Table 1. Our dataset covers eight languages with rich human–AI transitions. Language-centered statistics are reported in Appendix 8.

Table 1: Statistics of MuST-Mix and the five corpora used to construct MuST-Mix. #Samples is the number of paragraphs; average length counts the number of words in each paragraph; boundary counts the number of transitions in each paragraph.

Dataset	Language	Generator families	#Domains	Avg. length	#Samples	Boundary counts (0/1/2/3/4/5/6)
DetectRL-X	EN/ZH/DE/FR ES/RU/AR/PT	DeepSeek; GPT; Gemini; Qwen	6	557.2	324,603	10,301 / 37,038 / 143,225 / – / – / – / –
LLMTrace	EN/RU	GPT; Gemini; LLaMA; Phi; Qwen	9	257.3	727,849	73,195 / 123,784 / 166,926 / – / – / – / –
HC3	EN/ZH	GPT	7	168.8	98,355	11,826 / 25,391 / 13,573 / 2,143 / 973 / 69 / 27
C-ReD	ZH	Claude; DeepSeek; Doubao; GPT; Gemini; Qwen	5	525.6	163,001	1,167 / 23,778 / 62,058 / 10,519 / 14,293 / 1,223 / 965
M4GT Task 3	EN	GPT; LLaMA	2	157.1	37,495	1,157 / 8,713 / 8,447 / 2,701 / 1,258 / 170 / 51
MuST-Mix	EN/ZH/DE/FR ES/RU/AR/PT	Claude; DeepSeek; Doubao; GPT; Gemini; LLaMA; Phi; Qwen	20	352.5	1,351,303	97,646 / 218,704 / 394,229 / 15,363 / 16,524 / 1,462 / 1,043

4 METHOD

4.1 TASK DEFINITION

Fine-grained MGT detection involves two complementary tasks: span-level detection and boundary detection. As shown in Figure 1, relying on one task may introduce bias into the evaluation of fine-grained MGT detection performance. Hence, the evaluation necessitates the joint consideration of both tasks. Following prior work (Su et al., 2025), span-level detection (called word- or sentence-level detection in some literature) aims to assign each text span i a provenance label $y_i \in \{0,1\}$, with

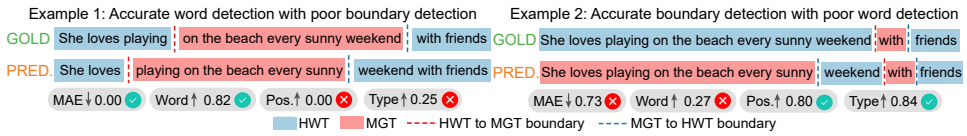


Figure 1: Two cases illustrate the complementary nature of span-level detection and boundary detection. Each example compares the gold and predicted labels. The definitions of four metrics are given in Section 4.1.

zero for human and one for AI. By aggregating the span-level labels, we can estimate the machine-generated proportion, or AI ratio, of a document: $r_{\text{AI}} = \frac{1}{N}\sum_{i=1}^{N} y_i$, where N is the number of text spans in the document. To measure the span-level detection performance, the mean absolute error (MAE) (Lei et al., 2025) of the AI ratio and the word-level F1 score ($F_{1,\text{word}}$) (Su et al., 2025) are introduced, as shown in Equation 2.

$$\text{MAE} = \frac{1}{D}\sum_{d=1}^{D} |\hat{r}_{\text{AI}}^{(d)} - r_{\text{AI}}^{(d)}|, \quad F_{1,\text{word}} = \frac{F_1^{\text{AI}} + F_1^{\text{human}}}{2} \tag{2}$$

where $r_{\text{AI}}^{(d)}$ is the ground-truth AI ratio of a document d, $\hat{r}_{\text{AI}}^{(d)}$ is the predicted value, D is the number of evaluated documents. $F_1^c = \frac{2\,\text{TP}_c}{2\,\text{TP}_c+\text{FP}_c+\text{FN}_c}$, and $c \in \{\text{human}, \text{AI}\}$ over the evaluation set.

Boundary detection, on the other hand, identifies where provenance changes and the direction of each transition (Zeng et al., 2024; Tolstykh et al., 2026a). Given sentence boundary j in paragraph d, let $b_j^{(d)}$ and $\hat{b}_j^{(d)}$ be the gold and predicted transition labels. The position F1 score and the F1 score of the boundary type are calculated as:

$$F_{1,\text{pos}} = \frac{2|B \cap \hat{B}|}{|B| + |\hat{B}|}, \qquad F_{1,\text{type}} = \frac{1}{3}\sum_{c\in\{0,1,2\}} \frac{2\,\text{TP}_c}{2\,\text{TP}_c + \text{FP}_c + \text{FN}_c}, \tag{3}$$

where $B = \{(d,j) : b_j^{(d)} \neq 0\}$ and $\hat{B} = \{(d,j) : \hat{b}_j^{(d)} \neq 0\}$. $c \in \{0,1,2\}$ denotes the three transition classes, i.e., no change, human-to-AI, and AI-to-human. And TP_c, FP_c, and FN_c count the sentence boundaries of a class c over the evaluation set.

MuST therefore addresses two complementary tasks by jointly optimizing span-level detection and boundary detection. Figure 2 summarizes the architecture consisting of three main components. First, the preprocessing module takes tokenized text as input and outputs the embeddings of each word, N-grams, sentences, and normalized positions. Second, the contextual encoder fuses the contextual information through a Transformer module to obtain contextual embeddings, which are used to augment the original word embeddings. Finally, we design a loss function combining the span-level detection and boundary detection tasks. Implementation details are shown in Appendix A.1.

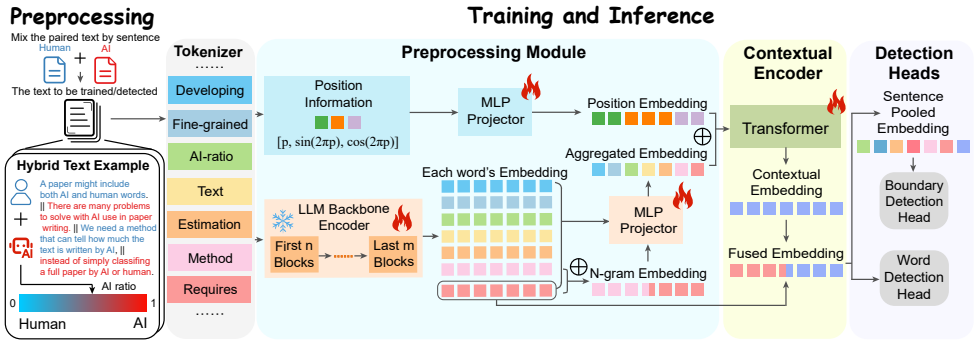


Figure 2: Overview of MuST. Given tokenized text, the preprocessing module forms word, N-gram, aggregated, and position embeddings, which contain different contextual information. Then, the Contextual Encoder derives contextual embeddings from aggregated embeddings and position embeddings, which are used to augment the word embeddings. Finally, the word detection head and boundary detection head make the final predictions.

4.2 Contextual Encoder

This module accepts word embedding, aggregated, and position embeddings as input, and it outputs a fused embedding for each word that includes rich contextual information.

Word embedding. The backbone LLM outputs embeddings for each word. Then we obtain aggregated embeddings by integrating the focal word's contextual information. Specifically, given a word i, let $\mathcal{T}_i = \text{Tokenizer}(\mathbf{x})$ be its subword tokens. $\mathbf{x}$ is the input document. The word embedding $\mathbf{e}_{\text{word},i}$ is calculated as:

$$\mathbf{e}_{\text{word},i} = \text{LLM}_{\text{ft}}(\mathcal{T}_i), \tag{4}$$

where $\mathrm{LLM}_{\mathrm{ft}}$ is the partially fine-tuned LLM that encodes the document, whose states over $\mathcal{T}_i$ are mean-pooled and projected into the model dimension d.

N-gram embedding. Inspired by NGM (Qu et al., 2026), to ensure that rarely occurring N-grams have an informative representation, we construct each N-gram embedding from the words it covers. An N-gram g is a sequence of contiguous words with start and end positions s_g and $e_g = s_g + |g| - 1$. The N-gram embedding $\mathbf{e}_{\mathrm{ngram},g}$ is calculated as:

$$\mathbf{e}_{\mathrm{ngram},g} = \mathrm{MeanPool}(\{\mathbf{e}_{\mathrm{word},i} : i \in \mathcal{I}_g\}), \tag{5}$$

where $\mathcal{I}_g = \{i : s_g \le i \le e_g\}$ is the contiguous set of word positions the N-gram spans, and $\mathrm{MeanPool}$ averages their word embeddings. As a result, the N-gram representation captures the distinguishable cues of multiple words.

Aggregated embedding. Inspired by prior work, we use adjacent words (Farzi & Farzi, 2026) and their sentence (Patel & Domeniconi, 2024) to enrich an n-gram embedding before sending it to the encoder. Specifically, given an N-gram g, let $\mathbf{e}_{\mathrm{ngram},g}$ be its N-gram embedding, $\mathbf{e}_{\mathrm{word},\ell_g}$ and $\mathbf{e}_{\mathrm{word},r_g}$ the word embeddings of its adjacent words, and $\mathbf{e}_{\mathrm{sent},c(g)}$ the embedding of its enclosing sentence. The aggregated embedding $\mathbf{e}_{\mathrm{aggregated},g}$ is calculated as:

$$\mathbf{e}_{\mathrm{aggregated},g} = \mathrm{MLP}_{\mathrm{ctx}}\big(\mathrm{LN}\big([\mathbf{e}_{\mathrm{ngram},g}; \mathbf{e}_{\mathrm{word},\ell_g}; \mathbf{e}_{\mathrm{word},r_g}; \mathbf{e}_{\mathrm{sent},c(g)}]\big)\big), \tag{6}$$

where $\ell_g = s_g - 1$ and $r_g = e_g + 1$ index the words adjacent to the N-gram. $\mathbf{e}_{\mathrm{sent},c(g)}$ is the mean-pooled word embedding of its enclosing sentence. LN is layer normalization, and $\mathrm{MLP}_{\mathrm{ctx}}$ is a two-layer feed-forward network $\mathbb{R}^{4d} \to \mathbb{R}^d \to \mathbb{R}^d$ with a GELU nonlinearity and dropout.

Position embedding. Prior studies find that sinusoidal position encodings are particularly useful for multilingual language models (Ravishankar & Søgaard, 2021). Hence, we follow this line of research and enrich each N-gram embedding with a position embedding. Given an N-gram g, let $\mathrm{pos}_g = s_g$ denote its start position. The position embedding $\mathbf{e}_{\mathrm{position},g}$ is calculated as:

$$\mathbf{e}_{\mathrm{position},g} = \mathrm{LN}\big(\mathrm{MLP}_{\mathrm{pos}}\big([\widehat{\mathrm{pos}}_g; \sin(2\pi\widehat{\mathrm{pos}}_g); \cos(2\pi\widehat{\mathrm{pos}}_g)]\big)\big), \tag{7}$$

where $\widehat{\mathrm{pos}}_g = \mathrm{pos}_g / L$, and L is the window length. $\mathrm{MLP}_{\mathrm{pos}}$ is a two-layer feed-forward network $\mathbb{R}^3 \to \mathbb{R}^d \to \mathbb{R}^d$ with a GELU nonlinearity, and LN is layer normalization, so one relative location is encoded identically in different contexts.

Contextual encoder. The goal of this step is to augment an n-gram representation with its neighboring n-grams. Given the ordered N-grams $\mathcal{G}_w$ in a window w, we stack their input embeddings into $\mathbf{E}_w = (\mathbf{e}_{\mathrm{position},g} + \mathbf{e}_{\mathrm{aggregated},g})_{g \in \mathcal{G}_w}$. The encoder returns one output vector for each N-gram,

$$\mathbf{e}_g = \mathrm{ContextualEncoder}(\mathbf{E}_w)_g, \tag{8}$$

where $\mathbf{E}_w$ is the input sequence of the window, $\mathbf{e}_g$ is the encoder output for N-gram g, and $\mathrm{ContextualEncoder}$ applies layer normalization and a Transformer encoder over the window.

Fused embedding. Our goal is to augment a word embedding with its N-gram information. Given a word i and the N-gram occurrences covering it, the contextual embedding $\mathbf{e}_{\mathrm{contextual},i}$ and the fused embedding $\mathbf{e}_{\mathrm{fused},i}$ are calculated as:

$$\mathbf{e}_{\mathrm{contextual},i} = \frac{1}{|\mathcal{G}(i)|} \sum_{g \in \mathcal{G}(i)} \mathbf{e}_g, \quad \mathbf{e}_{\mathrm{fused},i} = \mathbf{e}_{\mathrm{word},i} + \mathbf{e}_{\mathrm{contextual},i}. \tag{9}$$

where $\mathcal{G}(i)$ is the set of N-grams covering word i.

4.3 Detection Heads and Training Objective

Word detection head. The word detection head aims to decide the provenance of each word, based on which we assign span labels by majority voting. It takes the fused embeddings of a word as input and uses a linear layer with a sigmoid to map $\mathbf{e}_{\mathrm{fused},i}$ to $p_i \in [0, 1]$, the probability that word i is machine-generated.

Boundary detection head. The boundary head is designed to identify the point at which a text's provenance changes and to determine the direction of that transition. We mean-pool the fused embeddings of each word in a sentence j into $\mathbf{s}_j$ and compare it to adjacent sentences to detect the boundary transition point and direction,

$$\mathbf{q}_j = \mathrm{Softmax}(\mathrm{MLP}_{\mathrm{boundary}}[\mathbf{s}_{j-1}; \mathbf{s}_j; \mathbf{s}_j - \mathbf{s}_{j-1}]), \tag{10}$$

where $\mathbf{q}_j = \{q_{j,0}, q_{j,1}, q_{j,2}\}$ gives the probabilities of no change (0), human-to-AI (1), and AI-to-human (2), in that order.

Loss design. We design a loss function that jointly optimizes the two tasks. The model is optimized with cross-entropy for word provenance and transition type, together with a regularizer on the number of transitions,

$$\mathcal{L} = \mathcal{L}_{\text{word}} + \mathcal{L}_{\text{boundaryType}} + \lambda_{\text{boundaryCount}} \mathcal{L}_{\text{boundaryCount}}, \tag{11}$$

where $\mathcal{L}_{\text{word}}$ and $\mathcal{L}_{\text{boundaryType}}$ are the binary cross-entropy of the word detection head and the multiclass cross-entropy of the boundary detection head, $\mathcal{L}_{\text{boundaryCount}}$ is the Smooth L1 distance between the expected transition count $\sum_{j=2}^{N_{\text{span}}}(1-q_{j,0})$ and its ground-truth value, and $\lambda_{\text{boundaryCount}}$ weights that term. Parameter settings are reported in Appendix A.2.

5 Experiments

Our main experiment assesses the performance of MuST in fine-grained MGT detection across languages. We then perform ablation studies to analyze the effectiveness of each component and evaluate the robustness to unseen generators and domains. Finally, we analyze the performance of MuST under different attack settings. The runtime statistics are shown in Appendix A.10.

5.1 Experimental Setup

Dataset: We use MuST-Mix to evaluate MuST across eight languages. The eight languages share the same split protocol: 70/10/20 for train/validation/test. In terms of tokenization, one Chinese character forms a word span, while other languages use whitespace-delimited word spans.

Baselines. For the main comparisons, we compare MuST with fine-tuned detectors, statistics-based detectors, and span-level detectors. All supervised detectors are trained separately for each language and evaluated on the corresponding test set. Fine-tuned(FT) detectors comprise mDeBERTa-FT (Su et al., 2025; He et al., 2023), Qwen-FT (Su et al., 2025; Qwen Team, 2026), and XLM-RoBERTa-FT (Conneau et al., 2020). Statistics methods include GLTR (Gehrmann et al., 2019) and Fast-DetectGPT (Bao et al., 2024). The span-level detectors contain PTD (Li et al., 2024), GigaCheck (Tolstykh et al., 2026a), SenDetEX (Jiang et al., 2025), TriBERT (Zeng et al., 2024), Qwen-GBDT (Qwen Team, 2026; Friedman, 2001), and Phi-4-GBDT (Microsoft Research, 2025; Friedman, 2001). To evaluate the robustness under attack settings, we additionally include TASTE (Li et al., 2026), a document-level detector that shows strong generalization to adversarial scenarios. Detailed settings of baseline models are specified in Appendix A.5.

Metrics. To include measurements for both span-level detection and boundary detection, we report AI-ratio MAE (MAE), word-level F1 (Word), boundary-type macro-F1 (Type), and exact boundary-position F1 (Pos.), as defined in Section 4.1.

5.2 Detection across Languages

MuST outperforms all baselines on four metrics across eight languages (Table 2 and Table 10). Against the best baseline of each metric, MuST reduces the AI-ratio MAE by **58.43%** (0.0267→0.0111), and improves word-level F1 by **+2.59%** (0.9604→0.9853), boundary-type F1 by **+18.50%** (0.7269→0.8614), and position F1 by **+34.50%** (0.5928→0.7973). Compared to baselines, MuST achieves greater performance gains in specific languages. For example, MuST leads by **+131.24%** (0.3150→0.7284) on Russian in position F1, and by **+49.71%** (0.5639→0.8442), **+25.20%** (0.5826→0.7294), and **+18.48%** (0.6592→0.7810) on Arabic, German, and Portuguese. We note that prior work has strong performance in English. MuST shows a modest improvement of **+5.29%** (0.7965→0.8386) in position F1.

5.3 Ablation Studies

Table 3 shows how each component contributes to the performance. In general, all components contribute positively to the final results; this pattern holds across all languages, as shown in Table 3 and Table 11. Specifically, the sentence embedding shows the largest effect: its removal increases AI-ratio MAE by **+140.54%** (0.0111→0.0267) and reduces the Type F1 and position

Table 2: Overall averages for the eight languages. German, Russian, and Spanish are the three languages where MuST scores lowest and are reported here. Results of the other five languages are listed in Table 10.

Detector	Overall				German				Russian				Spanish			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0111**	**0.9853**	**0.8614**	**0.7973**	**0.0141**	**0.9800**	**0.8148**	**0.7294**	**0.0146**	**0.9808**	**0.8149**	**0.7284**	**0.0152**	**0.9810**	**0.8372**	**0.7625**
Fine-tuned detectors																
mDeBERTa-FT	0.1023	0.8467	0.4787	0.2718	0.1347	0.7920	0.4510	0.2377	0.1074	0.8260	0.4452	0.2229	0.1549	0.7807	0.4114	0.1928
Qwen-FT	0.0267	0.9604	0.7269	0.1663	0.0303	0.9514	0.6625	0.0914	0.0326	0.9565	0.6757	0.1031	0.0334	0.9528	0.6860	0.0639
XLM-RoBERTa-FT	0.0733	0.9041	0.5705	0.3955	0.1125	0.8566	0.4732	0.2711	0.0964	0.8725	0.4864	0.2825	0.0896	0.8864	0.5100	0.3203
Statistics-based detectors																
GLTR	0.2512	0.5050	0.3318	0.0538	0.2555	0.4450	0.3291	0.0336	0.2352	0.5210	0.3366	0.0609	0.2502	0.5315	0.3378	0.0801
Fast-DetectGPT	0.3317	0.5372	0.2958	0.1013	0.3482	0.5471	0.2816	0.0980	0.3247	0.5255	0.3173	0.1083	0.3057	0.5464	0.2979	0.1091
Span-level detectors																
PTD	0.0462	0.9203	0.7219	0.5928	0.0331	0.9487	0.7125	0.5826	0.1085	0.8065	0.5326	0.3150	0.0269	0.9603	0.7658	0.6586
GigaCheck	0.0627	0.9008	0.4377	0.1801	0.0539	0.9168	0.3913	0.1057	0.0673	0.8929	0.3744	0.0771	0.0934	0.8783	0.3655	0.0650
SenDetEX	0.1326	0.8042	0.4364	0.2181	0.1723	0.7404	0.4252	0.2019	0.1497	0.7648	0.4024	0.1663	0.1577	0.7790	0.4107	0.1909
TriBERT	0.4358	0.4656	–	0.2689	0.4613	0.4525	–	0.3303	0.4387	0.4589	–	0.0600	0.4154	0.4749	–	0.2296
Qwen-GBDT	0.1583	0.7649	0.4100	0.1863	0.1972	0.7040	0.3979	0.1675	0.1754	0.7182	0.3757	0.1359	0.2010	0.7242	0.3816	0.1603
Phi-4-GBDT	0.1460	0.7836	0.4233	0.2014	0.1848	0.7220	0.4153	0.1867	0.1619	0.7397	0.3900	0.1527	0.1768	0.7526	0.3980	0.1757

F1 by **5.33%** (0.8614→0.8155) and **8.57%** (0.7973→0.7290), respectively. We note that different components have diverse contributions to languages. For example, the impact of sentence embedding is most pronounced for Chinese, where the Type F1 and position F1 scores decrease by **15.21%** (0.9183→0.7786) and **23.49%** (0.8803→0.6735), respectively (Table 11). Russian and Spanish, on the other hand, rely heavily on contextual encoder, without which the AI-ratio MAE rises by **+84.93%** (0.0146→0.0270) on Russian and **+88.82%** (0.0152→0.0287) on Spanish, and the position F1 decreases by **10.43%** (0.7284→0.6524) and **5.81%** (0.7625→0.7182), respectively. The word detection head and boundary detection head both contribute substantially. With the word head disabled, AI-ratio MAE rises by **+2101.80%** (0.0111→0.2444) and word-level F1 drops by **43.14%** (0.9853→0.5602), while Type F1 and position F1 drop by **3.26%** (0.8614→0.8333) and **5.48%** (0.7973→0.7536). With the boundary head disabled, word-level F1 drops by **0.08%** (0.9853→0.9845) and Type F1 and position F1 by **2.36%** (0.8614→0.8411) and **3.51%** (0.7973→0.7693).

Table 3: Ablation studies on MuST-Mix. Overall averages the eight languages; German, Russian, and Spanish are the three languages where MuST scores lowest and are reported here, and the other five languages are listed in Table 11. The Full MuST row repeats the in-domain MuST rows, and the default-backbone row is that same configuration, so it repeats them. Rows are grouped by component and backbone axes. The best score within each block is bolded.

	Overall				German				Russian				Spanish			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
Components																
Full MuST	**0.0111**	**0.9853**	**0.8614**	**0.7973**	**0.0141**	**0.9800**	**0.8148**	**0.7294**	**0.0146**	**0.9808**	**0.8149**	**0.7284**	**0.0152**	**0.9810**	**0.8372**	**0.7625**
w/o contextual encoder	0.0202	0.9662	0.8337	0.7538	0.0211	0.9573	0.7670	0.6582	0.0270	0.9610	0.7643	0.6524	0.0287	0.9591	0.8074	0.7182
w/o N-gram embedding	0.0118	0.9845	0.8565	0.7901	0.0152	0.9782	0.8038	0.7132	0.0154	0.9807	0.8109	0.7232	0.0159	0.9805	0.8338	0.7572
w/o sentence embedding	0.0267	0.9593	0.8155	0.7290	0.0235	0.9593	0.7682	0.6598	0.0260	0.9618	0.7639	0.6522	0.0277	0.9612	0.8059	0.7160
w/o adj. word embedding	0.0118	0.9845	0.8562	0.7896	0.0151	0.9780	0.8030	0.7125	0.0156	0.9803	0.8078	0.7181	0.0160	0.9800	0.8326	0.7555
w/o position embedding	0.0119	0.9844	0.8558	0.7892	0.0154	0.9780	0.8012	0.7098	0.0157	0.9802	0.8089	0.7197	0.0161	0.9801	0.8312	0.7540
w/o word head	0.2444	0.5602	0.8333	0.7536	0.2453	0.5422	0.7873	0.6883	0.2469	0.5369	0.7831	0.6809	0.2526	0.5647	0.8069	0.7180
w/o boundary head	0.0118	0.9845	0.8411	0.7693	0.0149	0.9783	0.7918	0.6986	0.0157	0.9804	0.7943	0.7011	0.0156	0.9806	0.8141	0.7308
Backbone																
Qwen3.5-0.8B (default)	0.0111	0.9853	0.8614	0.7973	0.0141	0.9800	0.8148	0.7294	0.0146	0.9808	0.8149	0.7284	0.0152	0.9810	0.8372	0.7625
Qwen3.5-2B	0.0098	0.9873	0.8784	0.8223	0.0132	0.9810	0.8276	0.7487	0.0131	0.9838	0.8388	0.7639	0.0137	0.9830	0.8546	0.7882
Qwen3.5-4B	**0.0092**	**0.9881**	**0.8859**	**0.8333**	**0.0118**	**0.9833**	**0.8422**	**0.7697**	**0.0126**	**0.9845**	**0.8458**	**0.7741**	**0.0134**	0.9839	**0.8610**	**0.7972**
Llama3.2-1B	0.0098	0.9873	0.8789	0.8229	0.0128	0.9817	0.8299	0.7515	0.0127	0.9844	0.8433	0.7700	0.0135	**0.9842**	0.8590	0.7944
Phi-4-mini-3.8B	0.0115	0.9853	0.8626	0.7989	0.0146	0.9797	0.8092	0.7209	0.0150	0.9813	0.8180	0.7335	0.0155	0.9816	0.8433	0.7713

5.4 Generalization Studies

We evaluate the robustness to unseen generators and domains on the English, Chinese, and German MuST-Mix subsets. For each language, we employ both the leave-one-generator-out and leave-one-domain-out settings. That is, we train the models on all but one generator/domain, and evaluate them on the held-out generator/domain. We report fold averages within each language.

Table 4 reports the leave-one-generator-out results. MuST leads on four metrics in all three languages. Specifically, MuST reduces the AI-ratio MAE by **50.49%** (0.0307→0.0152), and improves word-level F1 by **+2.73%** (0.9535→0.9795), boundary-type F1 by **+17.28%** (0.7177→0.8417),

and position F1 by **+38.96%** (0.5531→0.7686). We observe that the span-level baselines show strong performance in English. However, the performance declines dramatically in Chinese and German. Taking the strongest baseline PTD as an example, its AI-ratio MAE rises by **+468.69%** (0.0214→0.1217) in Chinese and **+54.67%** (0.0214→0.0331) in German, indicating the value of MuST in multilingual settings.

Table 5 reports the leave-one-domain-out results. On average, MuST leads the best baseline in four metrics (declining the AI-ratio MAE by **49.42%** (0.0429→0.0217), and improving word-level F1 by **+3.93%** (0.9330→0.9697), boundary-type F1 by **+18.13%** (0.6717→0.7935), and position F1 by **+33.78%** (0.5225→0.6990)). However, under the English corpus, PTD leads MuST by **+0.11%** (0.6979→0.6987) on the F1 score. We note that the performance of PTD declines dramatically in Chinese (0.3723) and German (0.4964), and hypothesize that PTD may optimize for English only. Compared to generalization to generators, domain generalization is more challenging. We observe that word-level F1 **1.00%** (0.9795→0.9697), boundary-type F1 **5.73%** (0.8417→0.7935), and position F1 **9.06%** (0.7686→0.6990) all decrease. This suggests that domain differences shift word embeddings more substantially than generator differences, which poses greater challenges to span-level MGT detection. This observation aligns with that in DetectRL-X (Wu et al., 2026).

Table 4: MuST-Mix cross-generator results. Overall averages the three evaluated languages in each row. Each language reports the macro-average over four leave-one-generator-out folds of Appendix Tables 12–15, which report each unseen generator with the languages separate and pooled.

Detector	Overall				English				Chinese				German			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0152**	**0.9795**	**0.8417**	**0.7686**	**0.0191**	**0.9762**	**0.8523**	**0.7842**	**0.0084**	**0.9887**	**0.8942**	**0.8451**	**0.0181**	**0.9736**	**0.7787**	**0.6765**
Fine-tuned detectors																
mDeBERTa-FT	0.1321	0.8036	0.4543	0.2376	0.1130	0.8304	0.4631	0.2517	0.1126	0.8438	0.4671	0.2525	0.1708	0.7367	0.4326	0.2087
Qwen-FT	0.0307	0.9535	0.7177	0.2739	0.0301	0.9585	0.7476	0.1191	0.0228	0.9634	0.7806	0.6270	0.0392	0.9387	0.6248	0.0755
XLM-RoBERTa-FT	0.0700	0.9058	0.6190	0.4556	0.0693	0.9005	0.5437	0.3599	0.0266	0.9646	0.8462	0.7425	0.1141	0.8522	0.4671	0.2644
Span-level detectors																
PTD	0.0587	0.9045	0.6941	0.5531	0.0214	0.9727	0.8387	0.7658	0.1217	0.7928	0.5442	0.3304	0.0331	0.9480	0.6994	0.5631
GigaCheck	0.0625	0.8983	0.4427	0.1872	0.0434	0.9270	0.5452	0.3506	0.0844	0.8569	0.4179	0.1473	0.0598	0.9109	0.3651	0.0636
SenDetEX	0.1490	0.7813	0.4294	0.2085	0.1385	0.7948	0.4368	0.2192	0.1348	0.8164	0.4424	0.2240	0.1738	0.7327	0.4091	0.1822

Table 5: MuST-Mix cross-domain results. Overall averages the three evaluated languages in each row. Each language reports the macro-average over six leave-one-domain-out folds of Appendix Tables 16–21, which report each unseen domain with the languages separate and pooled.

Detector	Overall				English				Chinese				German			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0217**	**0.9697**	**0.7935**	**0.6990**	**0.0299**	**0.9605**	**0.7918**	0.6979	**0.0106**	**0.9851**	**0.8613**	**0.7972**	**0.0245**	**0.9634**	**0.7275**	**0.6020**
Fine-tuned detectors																
mDeBERTa-FT	0.1487	0.7975	0.4796	0.2796	0.1006	0.8490	0.5443	0.3617	0.2005	0.7693	0.4519	0.2466	0.1449	0.7743	0.4426	0.2306
Qwen-FT	0.0429	0.9330	0.6675	0.2449	0.0460	0.9306	0.6891	0.0936	0.0277	0.9570	0.7405	0.5722	0.0550	0.9114	0.5729	0.0688
XLM-RoBERTa-FT	0.0882	0.8937	0.6286	0.4725	0.1067	0.8726	0.5690	0.4058	0.0250	0.9683	0.8140	0.6912	0.1329	0.8401	0.5028	0.3206
Span-level detectors																
PTD	0.0666	0.8981	0.6717	0.5225	0.0494	0.9401	0.7916	**0.6987**	0.1006	0.8275	0.5718	0.3723	0.0498	0.9267	0.6516	0.4964
GigaCheck	0.0758	0.8810	0.4170	0.1525	0.0737	0.8956	0.4522	0.2133	0.0839	0.8527	0.4186	0.1513	0.0698	0.8947	0.3803	0.0929
SenDetEX	0.1503	0.7910	0.4632	0.2555	0.1105	0.8380	0.5232	0.3343	0.1941	0.7629	0.4233	0.2080	0.1463	0.7721	0.4430	0.2243

5.5 Robustness against attacks

In this study, we evaluate the robustness of MuST against different attacks. We use the annotation in DetectRL-X (Wu et al., 2026), where five attack types are recorded, including Expand, Polish, Condense, Back-translation, and Character substitution. Expand, Polish, and Condense indicate extending, rewriting, and shortening the MGT text, respectively. Back-translation translates the MGT text through another language, and Character substitution replaces individual characters with visually similar ones. Since the attack targets MGT only, we perform detection at the document level and report AI-ratio MAE and word F1 score.

Table 6 shows the average performance on English, French, German, and Chinese, where MuST is the strongest on both metrics under three types of attacks, i.e., Expand, Polish, and Condense. Notably, MuST is vulnerable to Back-translation and Character substitution attacks. Taking Back-translation as an example, MuST increases AI-ratio MAE by **+4944.00%** (0.0050→0.2522) and decreases word F1 by **31.64%** (0.9952→0.6803). We hypothesize that this degradation occurs because translation alters lexical usage and shifts the word-embedding distribution relative to clean training text. Similarly, MuST exhibits poor performance on Character substitution. The AI-ratio MAE is

raised by **+4666.00%** (0.0050→0.2383) and word F1 decreases by **27.00%** (0.9952→0.7265). We hypothesize that altering characters changes subword tokenization, based on which the word embeddings and contextual embeddings are constructed. As a result, MuST fails to estimate word-level provenance. In terms of specific languages, MuST shows more limitations. For example, TASTE is ahead of MuST in all five attacks on both metrics in English, surpassing **87.94%** (0.1247→0.0150) in AI-ratio MAE and **+14.77%** (0.8606→0.9877) in word-level F1 on average.

Table 6: Robustness under the five types of attacks. Detectors are trained per language on the clean training data. Each attacked test set rewrites the MGT documents of the evaluation pool with the named attack, while the HWT documents stay unchanged. Each cell reports MAE (↓) and Word (↑) separated by |, document-count weighted over the English, French, German, and Chinese subsets, the languages with strict attack evaluations. Clean is the unattacked reference. Per-language attack results are reported in Appendix Tables 22–25.

Attack types	**Clean**		**Expand**		**Polish**		**Condense**		**Backtrans.**		**Char sub.**	
Detector	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑
MuST	**0.0050**	**0.9952**	**0.0043**	**0.9966**	**0.0053**	**0.9949**	**0.0111**	**0.9903**	0.2522	0.6803	0.2383	0.7265
Multilingual document-level detectors												
TASTE	0.0552	0.9590	0.1485	0.8221	0.1637	0.8142	0.1920	0.7967	0.2512	0.7538	0.1696	0.8113
Fine-tuned detectors												
mDeBERTa-FT	0.1264	0.8772	0.0960	0.8841	0.1342	0.8704	0.1669	0.8476	0.3347	0.6157	0.3009	0.6938
Qwen-FT	0.0214	0.9816	0.0105	0.9891	0.0261	0.9777	0.0423	0.9674	0.2650	0.6803	0.1869	0.7988
XLM-RoBERTa-FT	0.0562	0.9432	0.0472	0.9585	0.0566	0.9436	0.0683	0.9273	0.2369	0.7108	0.1700	0.8299
Span-level detectors												
PTD	0.0242	0.9647	0.0467	0.9162	0.0233	0.9673	0.0293	0.9723	**0.1823**	**0.7719**	0.0786	0.9117
GigaCheck	0.0135	0.9764	0.0585	0.8866	0.0141	0.9774	0.0187	0.9854	0.3200	0.5903	**0.0256**	**0.9603**
SenDetEX	0.1756	0.8302	0.1106	0.8690	0.1825	0.8242	0.2288	0.7881	0.3574	0.5968	0.3607	0.6127

6 Discussion

Though MuST shows leading performance in multilingual span-level MGT detection, we acknowledge four limitations in terms of MuST-Mix, the study design, and the framework, and propose potential directions for improvement. *First*, based on semantic matching and controlled replacement, we construct human–AI hybrid text at the sentence level. We plan to explore novel techniques to construct word-level benchmarks where MGT and HWT are mixed within a sentence. *Second*, due to the scarcity of non-English datasets, the number of samples in MuST-Mix is unbalanced across languages. To solve this issue, we plan to focus on non-English HWT and augment MuST-Mix to achieve a more balanced distribution. *Third*, due to constraints on computational resources, we limit the multi-fold generalization evaluation to three languages, Chinese, English, and German. We plan to extend the generalization study to more languages in the future. *Fourth*, Section 5.5 shows that MuST is vulnerable to adversarial scenarios. Especially for English, MuST is led by the strongest baseline under all types of attacks. In future research, we plan to explore a novel span-level MGT detection method that is robust against adversarial attacks under multilingual settings.

7 Conclusions

We present MuST, a framework for multilingual fine-grained MGT detection. We design a novel loss function that jointly optimizes span-level detection and boundary detection tasks. Besides, to accommodate a diverse set of languages, MuST includes a contextual encoder that fuses each word embedding with adjacent words, N-grams, the sentence, and its normalized position. To assess our framework under multilingual settings, we then present MuST-Mix, a large-scale multilingual dataset containing human–AI hybrid text at the fine-grained level. Evaluation on eight languages shows that MuST surpasses 11 SOTA detectors in both span-level detection and boundary detection tasks. Our study further reveals the robustness of MuST to unseen generators and domains.

AI Use Statement

Polish Writing. AI assistants were used to polish the writing of this paper. All AI-assisted text was manually checked and proofread to ensure its authenticity.

Coding Assistance. AI assistants were used to assist with coding and debugging to realise our ideas in the modeling and running experiments. All AI-assisted code was manually checked and verified.

All authors are willing to take full responsibility for AI assistance in our work.

REFERENCES

Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, Jonibek Mansurov, Ekaterina Artemova, Vladislav Mikhailov, Rui Xing, Jiahui Geng, Hasan Iqbal, Zain Muhammad Mujahid, Tarek Mahmoud, Akim Tsvigun, Alham Fikri Aji, Artem Shelmanov, Nizar Habash, Iryna Gurevych, and Preslav Nakov. LLM-DetectAIve: a tool for fine-grained machine-generated text detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2024. doi: 10.18653/v1/2024.emnlp-demo.35.

Ekaterina Artemova, Jason S Lucas, Saranya Venkatraman, Jooyoung Lee, Sergei Tilga, Adaku Uchendu, and Vladislav Mikhailov. Beemo: Benchmark of expert-edited machine-generated outputs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.naacl-long.357.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *International Conference on Learning Representations*, 2024.

Guangsheng Bao, Lihua Rong, Yanbin Zhao, Qiji Zhou, and Yue Zhang. Decoupling content and expression: Two-dimensional detection of ai-generated text. 2025. URL `https://arxiv.org/abs/2503.00258`.

Maha Ben-fares, Urchade Zaratiana, Simon Hernandez, and Pierre Holat. FI group at SemEval-2024 task 8: A syntactically motivated architecture for multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024. doi: 10.18653/v1/2024.semeval-1.170.

J. Bevendorff et al. Overview of the "Voight-Kampff" generative AI authorship verification task at PAN and ELOQUENT 2025. In *26th Working Notes of the Conference and Labs of the Evaluation Forum, CLEF 2025*, 2025.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. doi: 10.18653/v1/2020.acl-main.747.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019. doi: 10.18653/v1/N19-1423.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. ZEN: Pre-training Chinese text encoder enhanced by n-gram representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020. doi: 10.18653/v1/2020.findings-emnlp.425.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. doi: 10.1609/aaai.v37i11.26501.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. doi: 10.18653/v1/2024.acl-long.674.

Pourya Farzi and Saeed Farzi. A novel semantic embedding using complex mapping. *Neural Processing Letters*, 2026. doi: 10.1007/s11063-026-11842-5.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 2001.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2019. doi: 10.18653/v1/P19-3019.

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Bedi. A survey on the possibilities & impossibilities of AI-generated text detection. *Transactions on Machine Learning Research*, 2023.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. URL https://arxiv.org/abs/2301.07597.

Hans W. A. Hanley and Zakir Durumeric. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2024. doi: 10.1609/icwsm.v18i1.31333.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing, 2023. URL https://arxiv.org/abs/2111.09543.

Yongxin He, Shan Zhang, Yixuan Cao, Lei Ma, and Ping Luo. Detree: Detecting human-ai collaborative texts via tree-structured hierarchical representation learning. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-4887.

Fred Heiding, Bruce Schneier, Arun Vishwanath, Jeremy Bernstein, and Peter S. Park. Devising and detecting phishing emails using large language models. *IEEE Access*, 2024. doi: 10.1109/ACCESS.2024.3375882.

Lei Jiang, Desheng Wu, and Xiaolong Zheng. SenDetEX: Sentence-level AI-generated text detection for human-AI hybrid content via style and context fusion. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.268.

Laida Kushnareva, Tatiana Gaintseva, Dmitry Abulkhanov, Kristian Kuznetsov, German Magai, Eduard Tulchinskii, Serguei Barannikov, Sergey Nikolenko, and Irina Piontkovskaya. Boundary detection in mixed AI-human texts. In *First Conference on Language Modeling*, 2024.

Eric Lei, Hsiang Hsu, and Chun-Fu Chen. Pald: Detection of text partially written by large language models. In *International Conference on Learning Representations*, 2025.

Yafu Li, Zhilin Wang, Leyang Cui, Wei Bi, Shuming Shi, and Yue Zhang. Spotting AI's touch: Identifying LLM-paraphrased spans in text. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. doi: 10.18653/v1/2024.findings-acl.423.

Yuanfan Li, Qi Zhou, and Zexuan Xie. Learning from dictionary: Enhancing robustness of machine-generated text detection in zero-shot language via adversarial training. In *The Fourteenth International Conference on Learning Representations*, 2026.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, and James Zou. Quantifying large language model usage in scientific papers. *Nature Human Behaviour*, 2025. doi: 10.1038/s41562-025-02273-8.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. 2023. URL https://arxiv.org/abs/2209.10655.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. doi: 10.18653/v1/2023.emnlp-main.616.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. MultiSocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.acl-long.36.

Samuel Mensah, Kai Sun, and Nikolaos Aletras. An empirical study on leveraging position embeddings for target-oriented opinion words extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. doi: 10.18653/v1/2021.emnlp-main.722.

Microsoft Research. Phi-4-Mini model card. Hugging Face, 2025. URL https://huggingface.co/microsoft/Phi-4-mini-instruct.

Sheshera Mysore, Debarati Das, Hancheng Cao, and Bahareh Sarrafzadeh. Prototypical human-AI collaboration behaviors from LLM-assisted writing in the wild. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.852.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. doi: 10.18653/v1/2023.findings-emnlp.97.

Raj Patel and Carlotta Domeniconi. Subword attention and post-processing for rare and unknown contextualized embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024. doi: 10.18653/v1/2024.findings-naacl.88.

Chenxi Qing, Junxi Wu, Zheng Liu, Yixiang Qiu, Hongyao Yu, Bin Chen, Hao Wu, and Shu-Tao Xia. C-ReD: A comprehensive Chinese benchmark for AI-generated text detection derived from real-world prompts. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. doi: 10.18653/v1/2026.findings-acl.2119.

Yuwen Qu, Wenhui Dong, Chenyang Si, and Caifeng Shan. Ngm: A plug-and-play training-free memory module for llms, 2026. URL https://arxiv.org/abs/2605.16893.

Qwen Team. Qwen3.5-0.8B model card. Hugging Face, 2026. URL https://huggingface.co/Qwen/Qwen3.5-0.8B.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Vinit Ravishankar and Anders Søgaard. The impact of positional encodings on multilingual compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. doi: 10.18653/v1/2021.emnlp-main.59.

Shoumik Saha and Soheil Feizi. Almost AI, almost human: The challenge of detecting AI-polished writing. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. doi: 10.18653/v1/2025.findings-acl.1303.

L. D. M. S. Sai Teja, N. Siva Gopala Krishna, Ufaq Khan, Muhammad Haris Khan, and Atul Mishra. DAMASHA: Detecting AI in mixed adversarial texts via segmentation with human-interpretable attribution. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026. doi: 10.18653/v1/2026.findings-eacl.326.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. Zen 2.0: Continue training and adaption for n-gram enhanced text encoders. 2021. URL https://arxiv.org/abs/2105.01279.

Michal Spiegel and Dominik Macko. KInIT at SemEval-2024 task 8: Fine-tuned LLMs for multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024. doi: 10.18653/v1/2024.semeval-1.84.

Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. HACo-det: A study towards fine-grained machine-generated text detection under human-AI coauthoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. doi: 10.18653/v1/2025.acl-long.1069.

Minh Ngoc Ta, Dong Cao Van, Duc-Anh Hoang, Minh Le-Anh, Truong Nguyen, My Anh Tran Nguyen, Yuxia Wang, Preslav Nakov, and Dinh Viet Sang. FAID: Fine-grained AI-generated text detection using multi-task auxiliary and multi-level contrastive learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.eacl-long.151.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, Aleksandr Gordeev, Vladimir Dokholyan, and Maksim Kuprashevich. GigaCheck: Detecting LLM-generated content via object-centric span localization. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026a. doi: 10.18653/v1/2026.findings-acl.213.

Irina Tolstykh, Aleksandra Tsybina, Sergey Yakubson, and Maksim Kuprashevich. Llmtrace: A corpus for classification and fine-grained localization of ai-written text, 2026b. URL `https://arxiv.org/abs/2509.21269`.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024a. doi: 10.18653/v1/2024.acl-long.218.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024b. doi: 10.18653/v1/2024.eacl-long.83.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 2025. doi: 10.1162/coli_a_00549.

Junchao Wu, Yefeng Liu, Chenyu Zhu, Hao Zhang, Zeyu Wu, Tianqi Shi, Yichao Du, Longyue Wang, Weihua Luo, Jinsong Su, and Derek F. Wong. DetectRL-X: Towards reliable multilingual and real-world LLM-generated text detection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.acl-long.1773.

Feng Xiong, Thanet Markchom, Ziwei Zheng, Subin Jung, Varun Ojha, and Huizhi Liang. NCL-UoR at SemEval-2024 task 8: Fine-tuning large language models for multigenerator, multidomain, and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024. doi: 10.18653/v1/2024.semeval-1.25.

Songlin Yang, Yikang Shen, Kaiyue Wen, Shawn Tan, Mayank Mishra, Liliang Ren, Rameswar Panda, and Yoon Kim. Path attention: Position encoding via accumulating householder transformations. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.52202/085713-2080.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. A survey on detection of LLMs-generated content. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. doi: 10.18653/v1/2024.findings-emnlp.572.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guanliang Chen. Towards automatic boundary detection for human-AI collaborative hybrid essay in education. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. doi: 10.1609/aaai.v38i20.30258.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. LLM-as-a-coauthor: Can mixed human-written and machine-generated text be detected? In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024a. doi: 10.18653/v1/2024.findings-naacl.29.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025. URL `https://arxiv.org/abs/2506.05176`.

Zhongping Zhang, Wenda Qin, and Bryan Plummer. Machine-generated text localization. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024b. doi: 10.18653/v1/2024.findings-acl.495.

A Appendix

A.1 Training and Inference Algorithms

Notation. A paragraph is a sequence of words u_i with provenance labels $y_i \in \{0, 1\}$, zero for human and one for AI, and each sentence boundary j carries a transition label $b_j \in \{\text{no change}, \text{human-to-AI}, \text{AI-to-human}\}$. The AI ratio of a paragraph is the fraction of its words that are labelled AI, $r_{\text{AI}} = \frac{1}{N}\sum_{i=1}^{N} y_i$, where N is the number of words in the paragraph (Equation 2). The word detection head maps the fused embedding of a word to $p_i \in [0, 1]$, the probability that the word is machine-generated, and the boundary detection head maps adjacent sentence embeddings to the three transition probabilities $\mathbf{q}_j$ (Equations 9 and 10). The reported metrics are AI-ratio MAE (MAE), word-level F1 (Word), boundary-type macro-F1 (Type), and exact boundary-position F1 (Pos.), all defined in Section 4.1. Every detector is trained and evaluated per language on the 70/10/20 split of Section 5; *in-domain* means training and testing on the same language, and a *fold* is one round of the leave-one-generator-out and leave-one-domain-out evaluations of Tables 4 and 5: one generator or domain is unseen during training, and the fold is scored on its documents.

Algorithm 1 Model training and inference

Input: paragraphs, words u_i with provenance labels y_i, transition labels b_j, and pretrained backbone parameters θ_0.

Output: word probabilities $\hat{p}_i$, boundary transition probabilities $\hat{\mathbf{q}}_j$, the paragraph segmented into labelled words, and the AI ratio $\hat{r}$.

1: Build the word, N-gram, sentence, and adjacent embeddings of every N-gram from the backbone states, and combine them into its aggregated embedding (Equation 6).

2: Add the position embedding to the aggregated embedding of each N-gram and feed the result to the contextual encoder, which relates the N-gram occurrences of a window to one another (Equation 8).

3: Average the contextual embeddings of the N-grams covering a word and add the result to its word embedding, giving the fused embedding $\mathbf{e}_{\text{fused},i}$ (Equation 9).

4: Predict the provenance of every word from $\mathbf{e}_{\text{fused},i}$ with the word detection head, and mean-pool the fused embeddings of each sentence to predict its boundary transition with the boundary detection head (Equation 10).

5: Optimize the word cross-entropy, the boundary-type cross-entropy, and the boundary-count Smooth L1 term (Equation 11).

6: Take the most probable transition class at each sentence boundary and split the paragraph at its nonzero predictions. Label the words of each predicted region by majority vote over their word probabilities obtained with the threshold in Table 7; break tied votes with the mean word probability of the region. Merge adjacent regions of the same provenance and recompute the transition directions from their labels. Return the fraction of words that fall in the merged AI-labelled regions as $\hat{r}$; when no transition survives, return the fraction of words that the word detection head labels AI.

A.2 Reproducibility: Configuration and Hyperparameters

Table 7 lists the settings shared by the runs in all languages. The values follow the current experiment configuration.

Span definition. We define spans as words or sentences for MGT annotation and evaluation. One Chinese character forms a word span, and the other languages use whitespace-delimited word spans (Section 5). Every word receives one annotation label and contributes one count to both the AI-ratio denominator and the detection metrics. Backbone subword states aligned with each rare word are mean-pooled to obtain the rare word's embedding. N-grams are enumerated over these spans inside the contextual encoder (Section 4.2).

Table 7: Configuration shared by the runs in all languages.

Setting	Value
Backbone	Qwen3.5-0.8B (shared multilingual backbone)
Backbone layout	24 blocks, 1,024-dim states
Fine-tuned backbone blocks	last 8 of 24
Supervised span unit	word and sentence
Window	2,048 backbone tokens, 1-sentence overlap
Projection dim m	256
Evaluated N-gram lengths	$\{2\}$
contextual encoder	2 layers, 8 heads, FFN $4\times$
Dropout	0.1
Precision	float16 with automatic mixed precision
Gradient checkpointing	on
Optimizer	AdamW
Learning rates for backbone, N-gram, and heads	5×10^{-6}, 10^{-4}, and 3×10^{-4}
Schedule	linear warmup (5% of updates), then linear decay
Weight decay	0.01
Gradient clipping	1.0
Batch size	4 per device on one GPU, no accumulation (effective batch 4)
Epochs	1 (complete training set)
Loss weights for the word and boundary-type terms	1.0 and 1.0
Boundary-count weight	0.02
Word class weights for HWT and MGT	1.0 and 1.0
MGT word threshold	0.5
Overlap penalty λ_{ov}	0.01

A.3 Semantic-Matching Sentence Replacement

Algorithm 2 Semantic-matching sentence replacement

Input: a paired human paragraph and its machine-generated rewrite, embedding encoder E, target replacement size $k \in \{1, 2, 3\}$, and seed r.

Output: the human source, the one-, two-, and three-sentence replacement conditions, and a fully machine-generated condition, with provenance and directional boundary labels.

1: Take a human paragraph and its machine-generated rewrite as a pair; the pairing by index is performed during data preparation.

2: For every retained paragraph pair, split both paragraphs into ordered sentence sequences.

3: Enumerate every contiguous group of one to three sentences on each side, giving the nine size combinations (h, a), and encode each group with E.

4: Run a monotonic one-use dynamic program over the two sequences, where a matched move costs $1 - \cos(E(H_{i:i+h}), E(M_{j:j+a})) + \lambda_{\text{ov}}(h + a - 2)$.

5: Backtrace the minimum-cost path, and mark as eligible the matched groups whose two sides are not textually identical and whose similarity reaches the configured floor.

6: For $k = 1, 2, 3$, shuffle the eligible groups under a per-pair seed and take the subset whose human-side sentence counts sum to exactly k; discard the pair when any k is unreachable.

7: In human order, emit the aligned AI group where a selected group falls and the human group otherwise; the endpoint conditions emit one side only.

8: Assign provenance labels to the emitted sentences, derive the boundary positions and their directions from the adjacent provenance labels, verify that the replaced human-sentence count equals k, and write the conditions.

A.4 Evaluation Dataset Statistics

Table 8: Language-level statistics of the MuST-Mix conditions. Each row aggregates all source datasets available for the language. A condition is one of the five paragraph types the construction produces: the human source, the one-, two-, and three-sentence replacements, and the fully machine-generated rewrite (Appendix A.3). Each sample is one reconstructed paragraph of a condition, pooled over the available training, validation, and test splits. Average length counts words. The boundary field reports the number of records at each human-to-AI or AI-to-human transition.

Language	Generators	Domains	#Samples	Avg. length	Boundary distribution
Arabic	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash	Academic; News; Novel; SEO; Webtext; Wiki	44,117	529.8	0: 1,232; 1: 6,464; 2: 17,549
English	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash; ChatGPT; GPT; Gemini; Qwen; LLaMA; Phi; GPT-4; LLaMA-chat	Academic; News; Novel; SEO; Webtext; Wiki; Finance; Open QA; Medicine; CS/AI Wikipedia; Reddit ELI5; Article; Factual; Paper abstract; Poetry; Question; Review; Short form; Story; Outfox; PeerRead	445,650	256.5	0: 18,967; 1: 86,487; 2: 120,826; 3: 4,844; 4: 2,231; 5: 239; 6: 78
French	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash	Academic; News; Novel; SEO; Webtext; Wiki	47,365	530.1	0: 1,162; 1: 4,607; 2: 22,402
German	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash	Academic; News; Novel; SEO; Webtext; Wiki	46,867	440.1	0: 2,142; 1: 4,359; 2: 21,174
Portuguese	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash	Academic; News; Novel; SEO; Webtext; Wiki	46,997	530.7	0: 1,225; 1: 5,173; 2: 21,407
Russian	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash; GPT; Gemini; Qwen; LLaMA; Phi	Academic; News; Novel; SEO; Webtext; Wiki; Article; Factual; Paper abstract; Poetry; Question; Review; Short form; Story	374,997	342.7	0: 14,219; 1: 67,156; 2: 105,463
Spanish	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash	Academic; News; Novel; SEO; Webtext; Wiki	46,661	479.1	0: 1,107; 1: 6,091; 2: 20,263
Chinese	GPT-4o; Qwen-Max; DeepSeek-V3; Gemini-2.5-Flash; GPT-3.5-Turbo; Claude-3.5-Haiku; DeepSeek-R1; Qwen2.5; Qwen3; Doubao-1.5-Pro	Academic; News; Novel; SEO; Webtext; Wiki; Q&A; Film review; Composition; Academic writing	209,865	636.7	0: 1,548; 1: 28,191; 2: 84,928; 3: 10,519; 4: 14,293; 5: 1,223; 6: 965

Section 5 evaluates detectors on the eight MuST-Mix languages. The language-level statistics, including named generators, domains, lengths, and boundary distributions, are summarized in Table 8.

A.5 Baseline Implementation Details

Table 9 summarizes the backbone and adaptation setup of every baseline in Table 2, in the same order and grouping. Supervised entries are trained per language on the corresponding training split with the same word-level annotation protocol and use a fixed 0.5 decision threshold unless noted; the statistics-based entries score text with a frozen scoring model and fit only a calibration layer on data that was not used to train the scoring model, so their scoring model is part of the method definition.

Table 9: Backbone and training configuration of the adapted baselines, grouped as in Table 2.

Method	Backbone or model	Training or adaptation
Fine-tuned detectors		
mDeBERTa-FT	mdeberta-v3-base (He et al., 2023)	Word-level token supervision.
Qwen-FT	Qwen3.5-0.8B (Qwen Team, 2026)	HACo-Det-style token classifier.
XLM-RoBERTa-FT	xlm-roberta-base (Conneau et al., 2020)	Word-level binary classifier.
Statistics-based detectors		
GLTR	Qwen3.5-0.8B (scoring model)	Frozen scoring model; next-token rank statistics mapped to labels by a calibration layer.
Fast-DetectGPT	Qwen3.5-0.8B (scoring model)	Frozen scoring model; conditional probability curvature mapped to labels by a calibration layer.
Span-level detectors		
PTD (Li et al., 2024)	Longformer	Sentence-level paraphrased-text detector.
GigaCheck	Ministral-3-8B-Instruct-2512	Detection fine-tuning on formatted data.
SenDetEX	Qwen3.5-0.8B + SenDetEX (Jiang et al., 2025)	Frozen proxy features.
TriBERT	paraphrase-multilingual-mpnet-base-v2 (Zeng et al., 2024)	Triplet fine-tuning over same-provenance sentences.
Phi-4-GBDT	phi-4-mini-instruct (Microsoft Research, 2025)	Frozen sentence-level NLL features.
Qwen-GBDT	Qwen3.5-0.8B	Frozen sentence-level NLL features.

A.6 MuST-Mix Per-Language Details

Table 10 groups the five MuST-Mix languages that Table 2 does not report. Each language block reports the detectors under the same in-domain protocol, restricted to one language.

Table 10: In-domain detection on the five MuST-Mix languages not reported in Table 2. The MuST row reports the full configuration, the same run as the Full MuST row of Table 11. Each language block reports MAE, Word, Type, and Pos. in the same order as the main table.

Detector	English				Chinese				Arabic				French				Portuguese			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0103**	**0.9872**	**0.8895**	**0.8386**	**0.0058**	**0.9928**	**0.9183**	**0.8803**	**0.0073**	**0.9890**	**0.8936**	**0.8442**	**0.0102**	**0.9869**	**0.8727**	**0.8136**	**0.0113**	**0.9845**	**0.8502**	**0.7810**
Fine-tuned detectors																				
mDeBERTa-FT	0.0796	0.8828	0.5188	0.3253	0.0737	0.9014	0.5315	0.3360	0.0569	0.9082	0.5409	0.3500	0.1068	0.8390	0.4562	0.2438	0.1048	0.8435	0.4746	0.2658
Qwen-FT	0.0241	0.9660	0.7809	0.1247	0.0186	0.9701	0.8085	0.6736	0.0240	0.9649	0.7689	0.1217	0.0252	0.9612	0.7168	0.0766	0.0254	0.9605	0.7159	0.0758
XLM-RoBERTa-FT	0.0632	0.9130	0.5671	0.3904	0.0112	0.9864	0.8963	0.8235	0.0445	0.9264	0.5961	0.4259	0.0775	0.9053	0.5319	0.3422	0.0915	0.8861	0.5032	0.3078
Statistics-based detectors																				
GLTR	0.2640	0.4779	0.3261	0.0454	0.2378	0.5395	0.3303	0.0490	0.2728	0.5215	0.3346	0.0567	0.2538	0.4769	0.3276	0.0454	0.2407	0.5269	0.3325	0.0595
Fast-DetectGPT	0.3546	0.5238	0.2726	0.0991	0.3555	0.5442	0.2949	0.0885	0.3213	0.4955	0.3365	0.1148	0.3206	0.5554	0.2770	0.0923	0.3230	0.5600	0.2883	0.1006
Span-level detectors																				
PTD	0.0155	0.9805	0.8600	0.7965	0.0795	0.8559	0.6139	0.4310	0.0638	0.8747	0.7038	0.5639	0.0179	0.9727	0.8199	0.7360	0.0244	0.9632	0.7667	0.6592
GigaCheck	0.0414	0.9269	0.5225	0.3277	0.0712	0.8644	0.4599	0.2198	0.0585	0.9033	0.5242	0.3117	0.0627	0.9100	0.4062	0.1253	0.0535	0.9135	0.4578	0.2087
SenDetEX	0.1275	0.8179	0.4522	0.2368	0.0859	0.8839	0.5011	0.2976	0.0957	0.8531	0.4648	0.2537	0.1366	0.7946	0.4134	0.1943	0.1351	0.7997	0.4215	0.2030
TriBERT	0.4477	0.4591	–	0.4278	0.4279	0.4842	–	0.3723	0.4142	0.4597	–	0.4373	0.4426	0.4738	–	0.2303	0.4385	0.4616	–	0.0633
Qwen-GBDT	0.1508	0.7802	0.4244	0.2032	0.0962	0.8679	0.4896	0.2820	0.1294	0.8012	0.4174	0.1964	0.1576	0.7608	0.3885	0.1645	0.1588	0.7629	0.4053	0.1806
Phi-4-GBDT	0.1453	0.7925	0.4347	0.2163	0.0950	0.8715	0.4941	0.2886	0.1207	0.8135	0.4299	0.2099	0.1378	0.7925	0.4100	0.1884	0.1455	0.7848	0.4147	0.1930

A.7 Component-Ablation Protocol

The component ablations disable one component of the model at a time. Every other setting (N-gram length n=2, the contextual encoder, the optimizer and learning rates, the batch composition) and the evaluation protocol of Section 4.1 are unchanged, so each variant differs from the full model only in the component it omits.

Contextual-encoder and embedding ablations. Three variants drop one term of the aggregated embedding (Equation 6): the N-gram embedding, the embeddings of the two adjacent words, or the embedding of the enclosing sentence. A fourth drops the position embedding from the input of the contextual encoder (Equation 8). Disabling the contextual encoder removes only the encoder: the contextual term of a word is then the mean of the N-gram embeddings covering it, added to its word embedding (Equation 9). The N-gram information therefore still reaches the word, but the occurrences in a window are no longer related to one another.

Backbone ablations. The language-model backbone is replaced, and the rest of the model is kept identical: the same N-gram embeddings, projection dimension, prediction heads, training recipe, and evaluation protocol.

Boundary-Head-only training and evaluation (no word head). With the word head disabled, the word-level BCE loss is removed from the objective. Since the word head is disabled, the model emits only the boundary transitions, and the boundary-type and boundary-count terms are trained as usual. At inference, the predicted transitions still partition the paragraph, but the word labels no longer come from majority voting over the word probabilities inside each region. The labels of words inside the predicted boundaries are determined by the directions of the preceding and following boundaries: for example, if the preceding boundary is human-to-AI and the following boundary is AI-to-human, then all words between the two boundaries are predicted as MGT. The AI ratio and word-level F1 are computed by comparing each word's predicted label with its ground-truth label.

Word-Head-only training and evaluation (no boundary head). With the boundary head disabled, the boundary-type and boundary-count losses are removed, and the model emits one provenance probability per word. At evaluation, the sentence labels are derived by majority vote over the word probabilities within each sentence, and the transition types are derived from the changes of these labels, which is the same word-to-sentence labelling rule used for the baselines' boundary metrics. The word labels from these votes, the sentence labels derived from them, and the AI-ratio rule then proceed on the derived transitions exactly as in the full configuration.

A.8 Ablation Results for Additional Languages

Table 11 reports the English, Chinese, Arabic, French, and Portuguese ablations. The Full MuST row repeats the MuST row of Table 10; the default-backbone row is that same configuration, so it repeats them.

Table 11: Ablation results on the five MuST-Mix languages not reported in Table 3; the Full MuST row repeats the in-domain MuST row of Table 10. The best score within each component or backbone block is bolded, and empty cells denote unavailable results.

	English				Chinese				Arabic				French				Portuguese			
	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
Components																				
Full MuST	**0.0103**	**0.9872**	**0.8895**	**0.8386**	**0.0058**	**0.9928**	**0.9183**	**0.8803**	**0.0073**	**0.9890**	**0.8936**	**0.8442**	**0.0102**	**0.9869**	**0.8727**	**0.8136**	**0.0113**	**0.9845**	**0.8502**	**0.7810**
w/o contextual encoder	0.0151	0.9703	0.8661	0.8039	0.0088	0.9788	0.9182	0.8635	0.0179	0.9706	0.8854	0.8317	0.0213	0.9668	0.8378	0.7620	0.0215	0.9656	0.8232	0.7405
w/o N-gram embedding	0.0105	0.9870	0.8845	0.8313	0.0063	0.9919	0.9126	0.8721	0.0084	0.9871	0.8878	0.8358	0.0108	0.9860	0.8708	0.8108	0.0117	0.9844	0.8476	0.7770
w/o sentence embedding	0.0207	0.9710	0.8637	0.8000	0.0573	0.9147	0.7786	0.6735	0.0171	0.9712	0.8852	0.8316	0.0203	0.9682	0.8388	0.7633	0.0208	0.9667	0.8201	0.7359
w/o adj. word embedding	0.0108	0.9868	0.8849	0.8318	0.0060	0.9924	0.9157	0.8764	0.0083	0.9877	0.8908	0.8399	0.0104	**0.9869**	0.8685	0.8073	0.0118	0.9842	0.8464	0.7750
w/o position embedding	0.0106	0.9868	0.8855	0.8327	0.0059	0.9926	0.9162	0.8771	0.0081	0.9878	0.8921	0.8419	0.0113	0.9854	0.8661	0.8039	0.0120	0.9839	0.8455	0.7741
w/o word head	0.2402	0.5556	0.8555	0.7897	0.2322	0.5832	0.9146	0.8565	0.2611	0.5827	0.8535	0.7856	0.2377	0.5595	0.8439	0.7711	0.2394	0.5569	0.8217	0.7388
w/o boundary head	0.0113	0.9859	0.8675	0.8082	0.0059	**0.9928**	0.9054	0.8619	0.0082	0.9877	0.8737	0.8160	0.0107	0.9863	0.8496	0.7814	0.0117	0.9843	0.8321	0.7563
Backbone																				
Qwen3.5-0.8B (default)	0.0103	0.9872	0.8895	0.8386	0.0058	0.9928	0.9183	0.8803	0.0073	0.9890	0.8936	0.8442	0.0102	0.9869	0.8727	0.8136	0.0113	0.9845	0.8502	0.7810
Qwen3.5-2B	0.0094	0.9884	0.8962	0.8486	0.0041	0.9948	0.9378	0.9089	0.0068	0.9905	0.9054	0.8615	0.0090	0.9888	0.8912	0.8408	0.0094	**0.9878**	0.8754	0.8175
Qwen3.5-4B	**0.0079**	**0.9902**	**0.9109**	**0.8702**	**0.0040**	**0.9951**	**0.9402**	**0.9124**	**0.0060**	**0.9912**	**0.9164**	**0.8778**	0.0087	**0.9893**	0.8940	**0.8454**	**0.0091**	0.9877	0.8763	0.8193
Llama3.2-1B	0.0082	0.9897	0.9075	0.8647	0.0060	0.9924	0.9191	0.8814	0.0070	0.9893	0.8996	0.8530	**0.0086**	0.9892	**0.8942**	0.8453	0.0097	0.9872	**0.8788**	**0.8226**
Phi-4-mini-3.8B	0.0097	0.9882	0.8987	0.8520	0.0076	0.9902	0.8993	0.8522	0.0084	0.9890	0.8977	0.8502	0.0107	0.9865	0.8738	0.8152	0.0109	0.9857	0.8604	0.7960

A.9 MuST-Mix Fold Details and Complete Result Tables

Tables 12 through 25 collect the result tables of this appendix: the per-generator folds of Table 4, the leave-one-domain-out folds of Table 5, and the attack results of Table 6. The per-generator folds disaggregate the leave-one-generator-out results by unseen generator, one table per fold: a fold trains on the records of the other three generators and tests on the documents of the generator its caption names. Each leave-one-domain-out table corresponds to one unseen MuST-Mix domain: the detector is trained on the other five domains and tested on that domain, with all four generators mixed in the test fold. Every table reports the English, Chinese, and German subsets separately, and the Overall column pools them. Per-generator scores are recomputed from the archived per-document predictions and document-count weighted to the corresponding column. Values follow the per-language word conventions of Table 2 and the same four metrics as Table 5.

Table 12: Leave-one-generator-out results for gpt-4o: each detector is trained on the records of the other three generators and tested on the gpt-4o documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0112**	**0.9850**	**0.8598**	**0.7950**	**0.0108**	**0.9863**	**0.8747**	**0.8170**	**0.0067**	**0.9911**	**0.9055**	**0.8616**	**0.0161**	**0.9775**	**0.7992**	**0.7063**
Fine-tuned detectors																
mDeBERTa-FT	0.1104	0.8358	0.4745	0.2663	0.1013	0.8473	0.4615	0.2517	0.0817	0.8881	0.5105	0.3097	0.1481	0.7720	0.4515	0.2373
Qwen-FT	0.0260	0.9599	0.7420	0.2931	0.0254	0.9635	0.7658	0.1269	0.0215	0.9658	0.8034	0.6644	0.0311	0.9504	0.6568	0.0865
XLM-RoBERTa-FT	0.0632	0.9164	0.6354	0.4807	0.0650	0.9080	0.5485	0.3659	0.0131	0.9826	0.8844	0.8037	0.1115	0.8584	0.4726	0.2714
Span-level detectors																
PTD	0.0553	0.9155	0.7039	0.5646	0.0145	0.9830	0.8608	0.7975	0.0963	0.8477	0.5464	0.3309	0.0301	0.9549	0.7216	0.5954
GigaCheck	0.0586	0.9061	0.4494	0.1990	0.0445	0.9249	0.5410	0.3417	0.0737	0.8784	0.4256	0.1673	0.0574	0.9150	0.3820	0.0884
SenDetEX	0.1456	0.7890	0.4301	0.2106	0.1384	0.7955	0.4248	0.2046	0.1354	0.8184	0.4416	0.2245	0.1630	0.7532	0.4238	0.2026

Table 13: Leave-one-generator-out results for qwen-max: each detector is trained on the records of the other three generators and tested on the qwen-max documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0230**	**0.9695**	**0.8079**	**0.7192**	0.0421	0.9478	0.7647	0.6563	**0.0060**	**0.9923**	**0.9115**	**0.8701**	**0.0211**	**0.9682**	**0.7467**	**0.6300**
Fine-tuned detectors																
mDeBERTa-FT	0.1495	0.7773	0.4306	0.2036	0.1336	0.8003	0.4399	0.2192	0.1081	0.8565	0.4662	0.2493	0.2070	0.6745	0.3854	0.1421
Qwen-FT	0.0368	0.9467	0.6838	0.2762	0.0454	0.9415	0.6628	0.0994	0.0175	0.9703	0.7992	0.6548	0.0478	0.9280	0.5886	0.0714
XLM-RoBERTa-FT	0.0631	0.9101	0.6266	0.4681	0.0726	0.8878	0.5285	0.3395	0.0130	0.9835	0.8750	0.7878	0.1038	0.8586	0.4745	0.2745
Span-level detectors																
PTD	0.0638	0.8972	0.6868	0.5418	**0.0379**	**0.9483**	**0.7863**	**0.6901**	0.0898	0.8460	0.5868	0.3929	0.0351	0.9454	0.6725	0.5256
GigaCheck	0.0590	0.9074	0.4410	0.1829	0.0484	0.9294	0.5164	0.3031	0.0710	0.8731	0.4611	0.2130	0.0576	0.9199	0.3456	0.0325
SenDetEX	0.1479	0.7800	0.4219	0.1972	0.1498	0.7807	0.4196	0.1947	0.1185	0.8349	0.4462	0.2270	0.1755	0.7241	0.3998	0.1696

Table 14: Leave-one-generator-out results for DeepSeek-V3: each detector is trained on the records of the other three generators and tested on the DeepSeek-V3 documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0153**	**0.9773**	**0.8307**	**0.7528**	**0.0142**	**0.9819**	**0.8720**	**0.8133**	**0.0135**	**0.9795**	**0.8519**	**0.7835**	**0.0181**	**0.9705**	**0.7689**	**0.6624**
Fine-tuned detectors																
mDeBERTa-FT	0.1442	0.7795	0.4508	0.2322	0.1257	0.8132	0.4660	0.2528	0.1460	0.7864	0.4533	0.2334	0.1609	0.7393	0.4334	0.2106
Qwen-FT	0.0339	0.9457	0.6998	0.2425	0.0279	0.9590	0.7578	0.1235	0.0310	0.9471	0.7269	0.5403	0.0425	0.9313	0.6156	0.0673
XLM-RoBERTa-FT	0.0877	0.8765	0.5821	0.3999	0.0752	0.8907	0.5367	0.3523	0.0641	0.9116	0.7588	0.6021	0.1232	0.8277	0.4531	0.2481
Span-level detectors																
PTD	0.0653	0.8929	0.6832	0.5390	0.0159	0.9787	0.8521	0.7860	0.1499	0.7489	0.5058	0.2779	0.0303	0.9506	0.6911	0.5523
GigaCheck	0.0503	0.9177	0.4468	0.1994	0.0361	0.9343	0.5605	0.3902	0.0604	0.8992	0.4265	0.1602	0.0543	0.9193	0.3540	0.0486
SenDetEX	0.1728	0.7332	0.4160	0.1915	0.1515	0.7645	0.4318	0.2119	0.1707	0.7518	0.4237	0.2012	0.1960	0.6837	0.3927	0.1615

Table 15: Leave-one-generator-out results for Gemini-2.5-Flash: each detector is trained on the records of the other three generators and tested on the Gemini-2.5-Flash documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0113**	**0.9863**	**0.8685**	**0.8075**	**0.0094**	**0.9888**	**0.8976**	**0.8500**	**0.0074**	**0.9919**	**0.9080**	**0.8651**	**0.0172**	**0.9782**	**0.7999**	**0.7073**
Fine-tuned detectors																
mDeBERTa-FT	0.1245	0.8220	0.4611	0.2484	0.0915	0.8608	0.4850	0.2832	0.1146	0.8444	0.4385	0.2177	0.1673	0.7610	0.4603	0.2448
Qwen-FT	0.0262	0.9617	0.7451	0.2849	0.0218	0.9698	0.8042	0.1264	0.0213	0.9702	0.7931	0.6487	0.0355	0.9450	0.6380	0.0770
XLM-RoBERTa-FT	0.0661	0.9201	0.6325	0.4746	0.0643	0.9154	0.5611	0.3820	0.0162	0.9805	0.8668	0.7763	0.1178	0.8640	0.4680	0.2634
Span-level detectors																
PTD	0.0839	0.8549	0.6970	0.5551	0.0172	0.9807	0.8557	0.7895	0.1509	0.7286	0.5376	0.3199	0.0367	0.9410	0.7124	0.5790
GigaCheck	0.0825	0.8614	0.4328	0.1664	0.0447	0.9192	0.5628	0.3675	0.1324	0.7766	0.3582	0.0487	0.0698	0.8893	0.3787	0.0848
SenDetEX	0.1298	0.8230	0.4497	0.2347	0.1144	0.8385	0.4710	0.2655	0.1144	0.8606	0.4579	0.2434	0.1605	0.7698	0.4203	0.1952

Table 16: Leave-one-domain-out results for the Academic domain: each detector is trained on the other five domains and tested on the Academic documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0122**	**0.9828**	**0.8667**	**0.8064**	**0.0118**	**0.9851**	**0.9062**	**0.8660**	**0.0131**	0.9809	0.8484	**0.7784**	**0.0117**	**0.9826**	**0.8461**	**0.7761**
Fine-tuned detectors																
mDeBERTa-FT	0.0617	0.9189	0.6128	0.4541	0.0666	0.9176	0.6625	0.5342	0.0497	0.9365	0.6205	0.4536	0.0691	0.9022	0.5558	0.3755
Qwen-FT	0.0290	0.9542	0.7454	0.2720	0.0330	0.9515	0.7881	0.1234	0.0277	0.9546	0.7603	0.5917	0.0264	0.9565	0.6882	0.0940
XLM-RoBERTa-FT	0.0336	0.9585	0.7486	0.6343	0.0494	0.9419	0.7109	0.5998	0.0134	**0.9838**	**0.8669**	0.7759	0.0384	0.9493	0.6656	0.5245
Span-level detectors																
PTD	0.0608	0.8989	0.7236	0.5979	0.0168	0.9803	0.8971	0.8538	0.1496	0.7398	0.4695	0.2236	0.0162	0.9760	0.8034	0.7150
GigaCheck	0.0857	0.8660	0.4312	0.1774	0.1056	0.8704	0.4893	0.2691	0.1009	0.8053	0.4159	0.1642	0.0506	0.9234	0.3892	0.1001
SenDetEX	0.0817	0.8869	0.5654	0.3865	0.0695	0.9106	0.6575	0.5240	0.0953	0.8660	0.5073	0.2967	0.0798	0.8849	0.5333	0.3420

Table 17: Leave-one-domain-out results for the News domain: each detector is trained on the other five domains and tested on the News documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0218**	**0.9689**	**0.7842**	**0.6861**	**0.0148**	**0.9797**	0.8217	0.7394	**0.0131**	**0.9809**	**0.8463**	**0.7764**	**0.0378**	**0.9457**	**0.6833**	**0.5408**
Fine-tuned detectors																
mDeBERTa-FT	0.1850	0.7638	0.4425	0.2518	0.0665	0.9026	0.5689	0.3912	0.3063	0.6553	0.3547	0.1476	0.1825	0.7328	0.4029	0.2157
Qwen-FT	0.0456	0.9286	0.6567	0.2335	0.0327	0.9526	0.7234	0.1209	0.0345	0.9431	0.6970	0.5161	0.0698	0.8896	0.5483	0.0620
XLM-RoBERTa-FT	0.0884	0.8891	0.5909	0.4197	0.0768	0.9010	0.5459	0.3651	0.0361	0.9515	0.7603	0.6069	0.1531	0.8140	0.4655	0.2860
Span-level detectors																
PTD	0.0525	0.9168	0.6651	0.5157	0.0168	0.9777	**0.8269**	**0.7491**	0.0903	0.8497	0.5315	0.3172	0.0505	0.9228	0.6364	0.4800
GigaCheck	0.0719	0.8889	0.4092	0.1439	0.0568	0.9177	0.3892	0.1107	0.0755	0.8720	0.4516	0.2008	0.0836	0.8765	0.3868	0.1200
SenDetEX	0.1583	0.7780	0.4444	0.2421	0.0781	0.8816	0.5310	0.3412	0.2332	0.7131	0.3677	0.1636	0.1640	0.7385	0.4340	0.2208

Table 18: Leave-one-domain-out results for the Novel domain: each detector is trained on the other five domains and tested on the Novel documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0176**	**0.9701**	**0.6654**	**0.5064**	**0.0218**	**0.9625**	**0.5566**	**0.3471**	**0.0044**	**0.9935**	**0.8765**	**0.8172**	**0.0263**	**0.9545**	**0.5666**	**0.3601**
Fine-tuned detectors																
mDeBERTa-FT	0.1179	0.8009	0.4154	0.1675	0.1388	0.7394	0.3663	0.0937	0.0503	0.9299	0.5327	0.3210	0.1635	0.7355	0.3490	0.0904
Qwen-FT	0.0433	0.9244	0.5408	0.1986	0.0480	0.9118	0.4668	0.0291	0.0206	0.9676	0.7407	0.5473	0.0611	0.8946	0.4181	0.0251
XLM-RoBERTa-FT	0.1318	0.8332	0.4979	0.2971	0.1572	0.7928	0.3373	0.0880	0.0109	0.9854	0.8381	0.7274	0.2254	0.7237	0.3238	0.0828
Span-level detectors																
PTD	0.0896	0.8579	0.5619	0.3546	0.1136	0.8596	0.5086	0.2777	0.0962	0.8213	0.7125	0.5702	0.0591	0.8925	0.4652	0.2167
GigaCheck	0.0808	0.8615	0.3821	0.0852	0.0480	0.9138	0.3506	0.0313	0.0762	0.8528	0.4407	0.1809	0.1185	0.8175	0.3562	0.0449
SenDetEX	0.1292	0.7997	0.3904	0.1388	0.1179	0.7933	0.3751	0.1069	0.0876	0.8681	0.4678	0.2303	0.1815	0.7387	0.3294	0.0806

Table 19: Leave-one-domain-out results for the SEO domain: each detector is trained on the other five domains and tested on the SEO documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0267**	**0.9682**	**0.8855**	**0.8349**	0.0609	**0.9278**	0.8648	0.8070	**0.0074**	**0.9920**	**0.9211**	**0.8856**	**0.0112**	**0.9854**	**0.8710**	**0.8127**
Fine-tuned detectors																
mDeBERTa-FT	0.1126	0.8500	0.5218	0.3396	0.1385	0.8090	0.5315	0.3552	0.1085	0.8587	0.4959	0.3031	0.0905	0.8829	0.5378	0.3600
Qwen-FT	0.0394	0.9478	0.7677	0.3017	**0.0608**	0.9231	0.7775	0.1263	0.0297	0.9647	0.8153	0.6764	0.0275	0.9558	0.7106	0.1072
XLM-RoBERTa-FT	0.0503	0.9419	0.7531	0.6405	0.0798	0.9069	0.7062	0.5917	0.0241	0.9741	0.8735	0.7828	0.0464	0.9453	0.6811	0.5487
Span-level detectors																
PTD	0.0555	0.9264	0.7807	0.6831	0.0667	0.9211	**0.8757**	**0.8231**	0.0753	0.8905	0.6731	0.5242	0.0245	0.9675	0.7929	0.7014
GigaCheck	0.0752	0.8919	0.4517	0.2136	0.0826	0.8855	0.5598	0.3956	0.0949	0.8656	0.3771	0.0858	0.0483	0.9244	0.4164	0.1565
SenDetEX	0.1315	0.8270	0.5020	0.3173	0.1663	0.7745	0.4945	0.3101	0.1344	0.8255	0.4719	0.2809	0.0935	0.8816	0.5395	0.3607

Table 20: Leave-one-domain-out results for the Webtext domain: each detector is trained on the other five domains and tested on the Webtext documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0295**	**0.9584**	**0.7741**	**0.6736**	**0.0480**	**0.9371**	0.7740	0.6781	**0.0131**	**0.9808**	**0.8338**	**0.7581**	**0.0282**	**0.9563**	**0.7145**	**0.5845**
Fine-tuned detectors																
mDeBERTa-FT	0.2408	0.7160	0.4520	0.2540	0.0920	0.8735	0.6003	0.4502	0.4649	0.5125	0.3420	0.1119	0.1587	0.7692	0.4204	0.2087
Qwen-FT	0.0393	0.9342	0.6793	0.2382	0.0516	0.9197	0.7068	0.0653	0.0244	0.9561	0.7459	0.5729	0.0425	0.9260	0.5864	0.0683
XLM-RoBERTa-FT	0.1578	0.8274	0.5664	0.3929	0.2124	0.7768	0.5142	0.3488	0.0334	0.9544	0.7775	0.6265	0.2304	0.7485	0.4048	0.2011
Span-level detectors																
PTD	0.0853	0.8818	0.6364	0.4764	0.0662	0.9227	**0.7932**	**0.7081**	0.0859	0.8555	0.5409	0.3304	0.1039	0.8672	0.5746	0.3903
GigaCheck	0.0768	0.8822	0.4255	0.1687	0.0992	0.8641	0.4463	0.2176	0.0749	0.8682	0.4514	0.2010	0.0572	0.9137	0.3796	0.0895
SenDetEX	0.2179	0.7299	0.4465	0.2463	0.1051	0.8568	0.5744	0.4197	0.3853	0.5850	0.3479	0.1271	0.1581	0.7537	0.4229	0.1996

Table 21: Leave-one-domain-out results for the Wiki domain: each detector is trained on the other five domains and tested on the Wiki documents of MuST-Mix. The Overall column pools the three subsets; English, Chinese, and German are reported separately.

	Overall				English				Chinese				German			
Detector	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑	MAE ↓	Word ↑	Type ↑	Pos. ↑
MuST	**0.0221**	**0.9696**	**0.7839**	**0.6846**	0.0222	0.9705	0.8273	0.7496	**0.0122**	**0.9826**	**0.8419**	**0.7678**	**0.0318**	**0.9559**	**0.6833**	**0.5375**
Fine-tuned detectors																
mDeBERTa-FT	0.1765	0.7325	0.4305	0.2070	0.1010	0.8518	0.5361	0.3457	0.2236	0.7228	0.3656	0.1421	0.2050	0.6232	0.3897	0.1334
Qwen-FT	0.0608	0.9087	0.6136	0.2264	0.0500	0.9247	0.6721	0.0968	0.0293	0.9559	0.6840	0.5285	0.1027	0.8461	0.4858	0.0561
XLM-RoBERTa-FT	0.0669	0.9122	0.6141	0.4494	0.0643	0.9164	0.5995	0.4413	0.0324	0.9608	0.7679	0.6279	0.1037	0.8599	0.4762	0.2807
Span-level detectors																
PTD	0.0556	0.9073	0.6629	0.5083	**0.0161**	**0.9792**	**0.8479**	**0.7806**	0.1061	0.8080	0.5031	0.2684	0.0448	0.9343	0.6372	0.4749
GigaCheck	0.0639	0.8957	0.4021	0.1257	0.0501	0.9219	0.4779	0.2557	0.0809	0.8520	0.3751	0.0752	0.0609	0.9129	0.3533	0.0464
SenDetEX	0.1852	0.7217	0.4277	0.1984	0.1261	0.8110	0.5068	0.3036	0.2289	0.7194	0.3770	0.1496	0.2009	0.6351	0.3992	0.1420

Table 22: Strict mixed-pool attack results on MuST-Mix English.

Attack types	Clean		Expand		Polish		Condense		Backtrans.		Char sub.	
Detector	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑
MuST	**0.0047**	0.9953	0.0040	0.9970	0.0058	0.9945	0.0106	0.9902	0.3847	0.5466	0.2186	0.7746
Multilingual document-level detectors												
TASTE	0.0052	**0.9967**	**0.0030**	**0.9979**	**0.0039**	**0.9976**	**0.0064**	**0.9963**	**0.0568**	**0.9500**	**0.0051**	**0.9966**
Fine-tuned detectors												
mDeBERTa-FT	0.1026	0.8945	0.0456	0.9404	0.1017	0.8985	0.1457	0.8667	0.3429	0.6351	0.2956	0.6890
Qwen-FT	0.0150	0.9863	0.0087	0.9923	0.0204	0.9818	0.0316	0.9737	0.3546	0.5933	0.0572	0.9480
XLM-RoBERTa-FT	0.0551	0.9395	0.0431	0.9605	0.0540	0.9427	0.0652	0.9266	0.1951	0.7962	0.1301	0.8611
Span-level detectors												
GigaCheck	0.0055	0.9946	0.0138	0.9739	0.0054	0.9948	0.0074	0.9925	0.4183	0.4821	0.0295	0.9710
SenDetEX	0.1844	0.8218	0.0752	0.8998	0.1983	0.8112	0.2691	0.7554	0.4017	0.5603	0.3443	0.6383
PTD	0.0097	0.9908	0.0097	0.9932	0.0101	0.9905	0.0112	0.9874	0.1510	0.8672	0.0261	0.9781

Table 23: Strict mixed-pool attack results on MuST-Mix French.

Attack types	Clean		Expand		Polish		Condense		Backtrans.		Char sub.	
Detector	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑
MuST	0.0060	**0.9939**	**0.0056**	**0.9951**	0.0060	**0.9939**	0.0086	0.9907	0.1095	0.8707	0.0682	0.9376
Multilingual document-level detectors												
TASTE	0.0559	0.9570	0.0548	0.9571	0.0519	0.9595	0.0585	0.9555	0.1234	0.8961	0.0550	0.9577
Fine-tuned detectors												
mDeBERTa-FT	0.5000	0.3289	0.1366	0.8266	0.1678	0.8308	0.1901	0.8200	0.2686	0.7067	0.3004	0.6871
Qwen-FT	0.0218	0.9809	0.0108	0.9885	0.0265	0.9770	0.0373	0.9702	0.1755	0.8016	0.1177	0.8916
XLM-RoBERTa-FT	0.0689	0.9316	0.0610	0.9446	0.0694	0.9316	0.0752	0.9177	0.1452	0.8365	0.2589	0.7339
Span-level detectors												
GigaCheck	**0.0053**	0.9925	0.0685	0.8669	**0.0053**	0.9932	**0.0078**	**0.9937**	0.3275	0.5803	**0.0127**	**0.9821**
SenDetEX	0.1921	0.8068	0.1413	0.8281	0.2007	0.7989	0.2289	0.7810	0.2839	0.6925	0.3364	0.6458
PTD	0.0097	0.9907	0.0289	0.9436	0.0097	0.9907	0.0105	0.9882	**0.0615**	**0.9199**	0.0890	0.9086

Table 24: Strict mixed-pool attack results on MuST-Mix German.

Attack types	Clean		Expand		Polish		Condense		Backtrans.		Char sub.	
Detector	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑
MuST	**0.0064**	**0.9938**	**0.0057**	**0.9957**	**0.0065**	**0.9936**	0.0167	**0.9854**	0.0186	0.9845	0.2122	0.8045
Multilingual document-level detectors												
TASTE	0.0635	0.9503	0.0272	0.9821	0.0497	0.9626	0.1034	0.9196	0.1015	0.9157	0.0527	0.9610
Fine-tuned detectors												
mDeBERTa-FT	0.0113	0.9895	0.1166	0.8677	0.1643	0.8417	0.2026	0.8084	0.2034	0.8060	0.3622	0.6246
Qwen-FT	0.0257	0.9779	0.0134	0.9859	0.0324	0.9727	0.0526	0.9590	0.0551	0.9538	0.1872	0.8228
XLM-RoBERTa-FT	0.0922	0.9085	0.0788	0.9327	0.0940	0.9069	0.1120	0.8795	0.1064	0.8958	0.2209	0.7852
Span-level detectors												
GigaCheck	0.0076	0.9936	0.0612	0.8864	0.0103	0.9918	0.0258	0.9824	0.0422	0.9648	**0.0061**	**0.9940**
SenDetEX	0.2154	0.7926	0.1475	0.8393	0.2228	0.7857	0.2705	0.7415	0.2399	0.7692	0.3536	0.6461
PTD	0.0143	0.9862	0.0349	0.9415	0.0144	0.9860	**0.0163**	0.9808	**0.0162**	**0.9847**	0.0668	0.9387

Table 25: Strict mixed-pool attack results on MuST-Mix Chinese.

Attack types	Clean		Expand		Polish		Condense		Backtrans.		Char sub.	
Detector	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑	MAE ↓	Word ↑
MuST	**0.0029**	**0.9977**	**0.0021**	**0.9985**	**0.0030**	**0.9976**	**0.0085**	**0.9947**	0.4966	0.3182	0.4549	0.3883
Multilingual document-level detectors												
TASTE	0.0964	0.9320	0.5092	0.3514	0.5495	0.3369	0.5997	0.3154	0.7232	0.2533	0.5656	0.3297
Fine-tuned detectors												
mDeBERTa-FT	0.0982	0.9140	0.0853	0.9017	0.1030	0.9107	0.1292	0.8955	0.5245	0.3139	0.2451	0.7747
Qwen-FT	0.0232	0.9815	0.0090	0.9896	0.0251	0.9795	0.0478	0.9669	**0.4754**	**0.3716**	0.3864	0.5317
XLM-RoBERTa-FT	0.0082	0.9937	0.0057	0.9962	0.0090	0.9932	0.0206	0.9856	0.5017	0.3135	0.0697	**0.9396**
Span-level detectors												
GigaCheck	0.0358	0.9244	0.0908	0.8187	0.0352	0.9296	0.0340	0.9731	0.4927	0.3332	**0.0544**	0.8937
SenDetEX	0.1099	0.9001	0.0782	0.9091	0.1079	0.9012	0.1463	0.8747	0.5046	0.3646	0.4086	0.5203
PTD	0.0630	0.8908	0.1136	0.7861	0.0590	0.9019	0.0793	0.9327	0.5016	0.3144	0.1326	0.8212

A.10 RUNTIME

Table 26 reports the training and inference cost of MuST and the baselines on the same subset of the English in-domain dataset. Training MuST takes 2h58m, 1.4 times less than GigaCheck (4h08m) and 1.5 times more than TASTE (1h58m), while XLM-RoBERTa-FT, Qwen-GBDT, and Phi-4-GBDT train in under 20 minutes, and the statistics-based detectors need no training. Inference is the smaller cost for MuST: 5m23s over the full evaluation pool, the fastest of the span-level detectors and within 7 seconds of XLM-RoBERTa-FT (5m16s), showing MuST is efficient at inference. However, MuST's training cost is higher than that of the fine-tuned and gradient-boosting baselines: 2h58m against 2h02m for Qwen-FT, the closest fine-tuned baseline, and 17m36s for XLM-RoBERTa-FT.

Table 26: Training and inference cost on the English in-domain split, in wall-clock time on a single GPU. Training uses the same per-language split as Table 2, and inference covers the full evaluation pool. A value of 0 marks a detector that requires no training. TASTE's inference is given per evaluation cell, averaged over the 24 cells of its attack evaluation.

Detector	Training	Inference	Total
MuST	2h58m	5m23s	3h03m
Multilingual document-level detectors			
TASTE	1h58m	1m58s	2h00m
Fine-tuned detectors			
mDeBERTa-FT	1h15m	20m55s	1h36m
Qwen-FT	2h02m	11m23s	2h13m
XLM-RoBERTa-FT	17m36s	5m16s	22m52s
Statistics-based detectors			
GLTR	0	45m19s	45m19s
Fast-DetectGPT	0	4m06s	4m06s
Span-level detectors			
PTD	1h13m	6m04s	1h19m
GigaCheck	4h08m	29m42s	4h37m
SenDetEX	49m10s	59m57s	1h49m
Qwen-GBDT	18m16s	29m06s	47m22s
Phi-4-GBDT	14m30s	19m32s	34m02s