WETBench: A Benchmark for Detecting Task-Specific Machine-Generated Text on Wikipedia

Gerrit Quaremba[1], **Elizabeth Black**[1], **Denny Vrandečić**[2], **Elena Simperl**[1]
[1]King's College London, [2]Wikimedia Foundation
{gerrit.quaremba,elizabeth.black,elena.simperl}@kcl.ac.uk
denny@wikimedia.org

Abstract

Given Wikipedia's role as a trusted source of high-quality, reliable content, concerns are growing about the proliferation of low-quality machine-generated text (MGT) produced by large language models (LLMs) on its platform. Reliable detection of MGT is therefore essential. However, existing work primarily evaluates MGT detectors on generic generation tasks rather than on tasks more commonly performed by Wikipedia editors. This misalignment can lead to poor generalisability when applied in real-world Wikipedia contexts. We introduce **WETBench**, a multilingual, multi-generator, and *task-specific* benchmark for MGT detection. We define three editing tasks, empirically grounded in Wikipedia editors' perceived use cases for LLM-assisted editing: *Paragraph Writing*, *Summarisation*, and *Text Style Transfer*, which we implement using two new datasets across three languages. For each writing task, we evaluate three prompts, generate MGT across multiple generators using the best-performing prompt, and benchmark diverse detectors. We find that, across settings, training-based detectors achieve an average accuracy of 78%, while zero-shot detectors average 58%. These results show that detectors struggle with MGT in realistic generation scenarios and underscore the importance of evaluating such models on diverse, task-specific data to assess their reliability in editor-driven contexts.

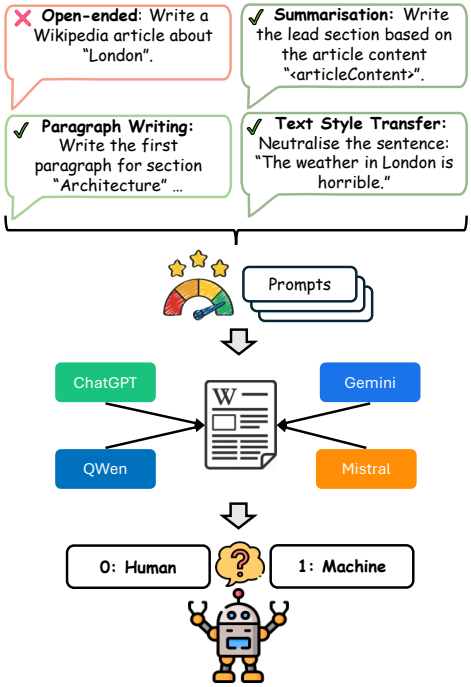

Figure 1: We define *task-specific editing scenarios* on Wikipedia, test various prompting techniques, generate LLM-written text using the best-performing prompts, and benchmark SOTA detectors on these data. This contrasts with prior work, which primarily focuses on a single, *open-ended* generation task that only partially captures the real-world editorial use of LLMs.

1 Introduction

Wikipedia serves as a vital source of high-quality, trustworthy data across artificial intelligence (AI) communities. Its scale and richness have played a foundational role in the development of large language models (LLMs) (Deckelmann, 2023; Longpre et al., 2023). However, the Wikipedia community has expressed growing concern about the increasing prevalence of machine-generated text (MGT) produced by LLMs on its platform.[1] The Wikimedia Foundation warns that the spread of low-quality, unreliable MGT in its projects could undermine its knowledge integrity.[2] Specifically, unverified MGT poses challenges such as factual fabrication (Huang et al., 2025a) and the perpetuation of biases present in training data (Gallegos et al., 2024), both of which jeop-

[1]https://en.wikipedia.org/wiki/Wikipedia:Large_language_models
[2]Wikipedia Community Call Notes 2023–24

Proceedings of the 2nd Workshop on Advancing Natural Language Processing for Wikipedia (WikiNLP 2025), pages 10–30
August 1, 2025

ardise Wikipedia's core content policies.[3] Additionally, given Wikipedia's frequent inclusion in LLM training corpora, undetected MGT on the platform may contribute to performance degradation in future models (Shumailov et al., 2024). Consequently, distinguishing human-written from machine-generated text has become increasingly important, leading to community efforts to identify and remove MGT,[4] and to a growing body of research on estimating the prevalence of MGT on Wikipedia (Brooks et al., 2024; Huang et al., 2025b).

Prior work on benchmarking MGT detectors (e.g., Guo et al., 2023; Li et al., 2023; Wang et al., 2023, 2024a) has included the Wikipedia domain but typically fails to reflect the complexities of editor-driven MGT instances on the platform. Existing experimental setups generally assume that MGT on Wikipedia results from (*i*) open-ended, topic-to-text generation and (*ii*) simplistic prompting techniques. These setups usually rely on a single prompt to generate an entire article, which diverges significantly from real-world Wikipedia editing practices that are task-specific and incremental. In fact, prompting an LLM to verbatim *"Write a Wikipedia article about [...]*," as done in earlier work, is explicitly discouraged by the community.[5]

These limitations in existing setups may obscure the actual performance of state-of-the-art (SOTA) detectors when applied to real-world Wikipedia contexts. Figure 2 shows that the textual characteristics of task-specific MGT—unlike open-ended, topic-to-text MGT—more closely resemble their human-written text (HWT) references. Detectors trained and evaluated on generic generation tasks may learn high-level textual patterns that are less transferable to task-specific MGT instances. Consequently, detectors may not generalise well to detecting diverse, task-specific MGT on Wikipedia, leaving an unknown number of instances with potentially harmful characteristics—such as hallucination or bias—largely undetected. To address this issue, we advocate for evaluating detectors on data that reflect practical use cases of editors integrating LLMs into their editorial workflows. This is essential for understanding the capacity of automatic detection methods to safeguard Wikipedia's knowledge integrity and to assist editors in identifying and removing low-quality MGT.

To this end, we build an MGT detection benchmark for *task-specific editing* scenarios on Wikipedia. To create our benchmark, we construct and release two new Wikipedia text corpora covering three languages with varying resource availability, enabling conclusions beyond the predominantly studied English Wikipedia. We then propose three editing tasks—*Paragraph Writing*, *Summarisation*, and *Text Style Transfer*—grounded in practical use cases identified by Ford et al. (2023), who analysed Wikipedia editors' perceived opportunities for LLM-assisted editing. For each task, we test various prompting techniques, generate MGT using diverse LLMs, and benchmark SOTA detectors across languages, generators, and tasks (see Figure 1). We hope that our multipurpose datasets will benefit the broader Wikipedia and AI communities in areas such as multilingual bias detection and single-document summarisation. We further aim to offer insights into the feasibility and reliability of automated detection methods for identifying MGT on Wikipedia.

Our contributions are as follows:

- We build two datasets for our benchmark covering English, Portuguese, and Vietnamese: **WikiPS**, a large-scale collection of high-quality (*i*) lead–infobox–body triplets and (*ii*) paragraphs; and **mWNC**, an extension of the WNC (Pryzant et al., 2020) to Portuguese and Vietnamese, and one of the first to include paragraph-level pairs for English.

- **W**ikipedia **E**diting **T**asks **B**enchmark, a comprehensive benchmark of 101,940 *task-specific* human-written and machine-generated Wikipedia texts, comprising three languages with varying levels of resource availability, four generators from two model families, and eight SOTA detectors from three detection families. We release all data and code on GitHub and plan to extend the benchmark with additional tasks, languages, and generators.

- We benchmark SOTA detectors on our data and find that detectors across all families struggle across tasks. While training-based detectors consistently outperform zero-shot meth-

[3] https://en.wikipedia.org/wiki/Wikipedia:Core_content_policies

[4] https://en.wikipedia.org/wiki/Wikipedia:WikiProject_AI_Cleanup

[5] https://en.wikipedia.org/wiki/Wikipedia:Large_language_models

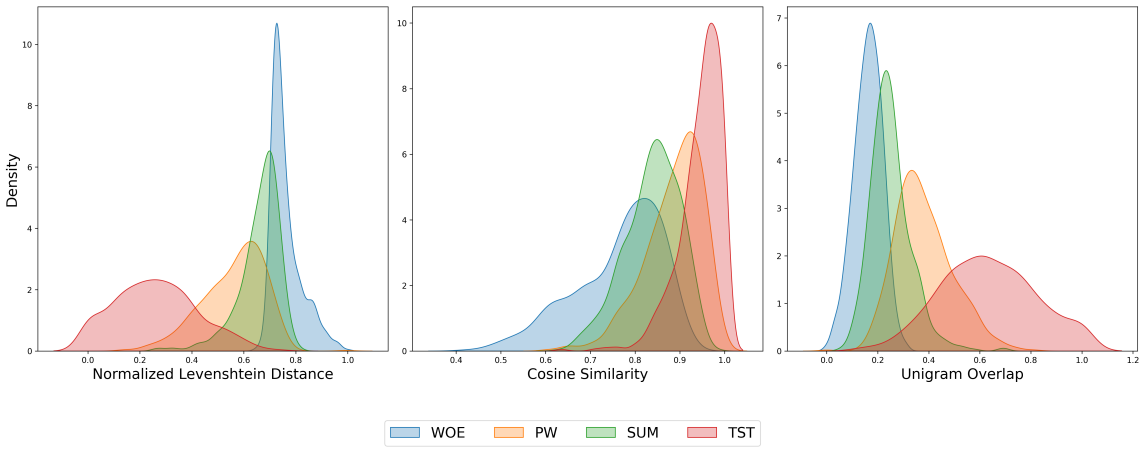

Figure 2: Comparison of MGT and HWT (N=600) for English Wikipedia Open-Ended Generation (WOE) vs. our Wikipedia editing tasks: Paragraph Writing (PW), Summarisation (SUM), and Text Style Transfer (TST). Task-specific MGT consistently demonstrates closer proximity to human writing across all dimensions.

ods, we observe substantial performance variation across languages, generators, and tasks.

2 Related Work

Wikipedia Editing Tasks We concentrate on three common editing tasks with varying degrees of LLM involvement: Paragraph Writing, Summarisation, and Text Style Transfer.

Paragraph Writing Generating new, encyclopaedic content—such as full paragraphs—is central to expanding knowledge on Wikipedia. This includes writing paragraphs from scratch, expanding article stubs, or rewriting existing content. With nearly half of all Wikipedia articles classified as stubs, researchers have extensively studied Wikipedia content generation.[6] The scope of generated content varies from paragraph-level (e.g., Liu et al., 2018; Balepur et al., 2023; Qian et al., 2023) to full-article generation (e.g., Sauper and Barzilay, 2009; Banerjee and Mitra, 2015; Fan and Gardent, 2022; Shao et al., 2024; Zhang et al., 2025). The methods employed range from early template-based approaches (Sauper and Barzilay, 2009) to more recent work using retrieval-augmented generation (RAG) with pre-trained language models (PLMs) (Fan and Gardent, 2022) or LLMs (Shao et al., 2024; Zhang et al., 2025).

Summarisation According to Wikipedia's Manual of Style,[7] each article should begin with a lead section that serves as an introduction by summarising its most important points. The literature treats lead section generation either as a multi-document (e.g., Liu et al., 2018; Ghalandari et al., 2020; Hayashi et al., 2021) or single-document (e.g., Casola et al., 2021; Gao et al., 2021; Perez-Beltrachini and Lapata, 2022; Sakota et al., 2023) summarisation problem. A model's objective is typically abstractive summarisation, that is, generating a lead section from scratch based on the article body.

Text Style Transfer Maintaining a Neutral Point of View[8] (NPOV) is a core Wikipedia policy, which states that all content must be written from a perspective that is fair, proportionate, and, as far as possible, free from editorial bias. Pryzant et al. (2020) introduce the Wikipedia Neutrality Corpus (WNC), a large-scale parallel corpus of biased and neutralised sentence pairs retrieved from NPOV-related revisions. They further introduce the task of *neutralisation*, a text style transfer task that aims to reduce subjectivity in a sentence while preserving its meaning. Recent work has used the WNC to improve data quality (Zhong et al., 2021), test generalisation to other domains (Salas-Jimenez et al., 2024), or examine the ability of LLMs to detect and neutralise bias (Ashkinaze et al., 2024).

MGT Detection Benchmarks There has been extensive work on benchmarking SOTA MGT detectors across diverse domains, languages, and generators. TuringBench (Uchendu et al., 2021) is one of the first benchmarks to study the Turing test and authorship attribution, using multiple generators in the news domain. MULTITuDE (Macko

[6] https://en.wikipedia.org/wiki/Wikipedia:Stub

[7] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

[8] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

et al., 2023) expands MGT data for languages other than English, testing detectors in multilingual settings. MAGE (Li et al., 2023) covers multiple domains, generators, and detectors, benchmarked across eight increasingly challenging detection scenarios. M4 (Wang et al., 2023) comprehensively includes various generators, languages, and domains, while M4GT (Wang et al., 2024b) expands on M4 by incorporating additional languages and introducing human-machine mixed detection. A recent line of work focuses on evading detectors through adversarial attacks (e.g., He et al., 2024; Wu et al., 2024; Zheng et al., 2025).

Most prior work has treated MGT generation primarily (*i*) as an open-ended text task, (*ii*) left different prompting techniques unexplored, and (*iii*) produced full articles with a single prompt. CUDRT (Tao et al., 2024) is a notable exception addressing (*i*) by introducing a bilingual, multi-domain benchmark that covers five types of LLM operations. However, it does not consider Wikipedia, lacks analysis of how different prompting techniques affect these operations, and is limited to only three detectors.

3 Dataset Construction

We construct two corpora for three languages with varying resource levels: English (high), Portuguese (medium), and Vietnamese (low). **WikiPS** includes paragraphs and lead–content pairs. **mWNC** is a multilingual version of the WNC (Pryzant et al., 2020). Appendix A provides detailed descriptions of the dataset construction, and Appendix Table 6 presents dataset statistics.

3.1 WikiPS

We construct **Wiki**pedia **P**aragraphs and **S**ummarisation, a large-scale collection of Wikipedia paragraphs and lead–content pairs. To ensure that our data is not contaminated by MGT, we use the latest versions of all mainspace articles prior to the release of ChatGPT on 30 November 2022. For each language, we randomly retrieve 100,000 non-stub articles from the MediaWiki Action API,[9] apply extensive filtering and cleaning of the HTML, and parse the lead section, infobox, paragraphs, and references. This forms our article-level base sample, from which we construct the paragraph and summarisation subsets, respectively.

[9] https://www.mediawiki.org/wiki/API:Main_page

Paragraphs For each language, we consider all paragraphs from 20,000 articles in our base sample. To ensure paragraph quality, we retain only those that contain at least three sentences and 20 characters, include at least one reference, and have word counts within two standard deviations of the respective sample mean. We also add diverse metadata, such as the paragraph's location on the page, to enable filtering for specific types of paragraphs.

Summarisation We retrieve lead–infobox–body triplets from all articles in each language, as information in the lead section is often sourced from the infobox (Gao et al., 2021). If an infobox is not available, we still extract the article, leaving the infobox field empty. We then merge the infobox (if present) and article body with minimal formatting into lead–content pairs. For English and Portuguese, we exclude pairs in which the lead/content is shorter than 10/100 characters, respectively, or longer than two standard deviations above the sample mean. For Vietnamese, we adjust the upper context limit to a minimum of 2,900 words due to its considerably longer articles. Appendix Table 7 compares our dataset to commonly used summarisation datasets.

3.2 mWNC

multilingual **WNC** extends the original WNC (Pryzant et al., 2020), which consists of English biased–neutralised sentence pairs, by adding pairs for Portuguese and Vietnamese, as well as paragraph-level pairs for English. We primarily follow the methodology of Pryzant et al. (2020), including crawling NPOV-related revisions, aligning pre- and post-neutralisation sentences, and applying rule-based filtering to improve precision. However, we modify their procedure by relaxing certain constraints to increase the number of instances for the Vietnamese Wikipedia, where the number of NPOV-related revisions is comparatively low. Furthermore, we are among the first to collect biased–neutralised paragraph-level pairs. We identify biased–neutralised paragraph pairs if three or more adjacent sentences each contain at least one NPOV-related edit. Due to the considerably smaller number of NPOV-related revisions in the other languages, we were only able to produce paragraph-level data for English.

4 Editing Tasks Design

We define three editing tasks with varying degrees of LLM intervention: *Paragraph Writing*, *Summarisation*, and *Text Style Transfer*. These tasks are empirically motivated by Ford et al. (2023), who found that Wikipedia editors see potential in LLMs for *generating article drafts or stubs*, *summarising content*, and *improving language*. We implement Paragraph Writing and Summarisation using the WikiPS corpus, and Text Style Transfer using the mWNC.

For each task and language, we evaluate three prompting strategies on a length-stratified 10% sample of the target data using GPT-4o mini,[10] and select the best-performing prompt to generate MGT for our benchmark. Appendix B provides implementation details and prompt templates.

4.1 Paragraph Writing

We define *Paragraph Writing* as the task of writing the opening paragraph of a new section, resembling a scenario in which an editor aims to add new content to an article. In contrast to prior work on open-ended generation (e.g., Guo et al., 2023; Li et al., 2023; Wang et al., 2023, 2024a), we frame this as a *content-conditioned* generation task, where the model receives additional information about the content and style of the output. This *content creation* task involves the highest degree of LLM contribution, as the model generates the paragraph from scratch.

We devise three prompts with increasing levels of content conditioning. **Minimal** simply instructs the model to write a paragraph given article and section titles. We include this prompt as it reflects generation settings in prior work and thus serves as a comparative baseline. **Content Prompts** expand Minimal by incorporating up to ten content prompts about the target HWT paragraph (e.g., "What is London's population?"), obtained from GPT-4o,[11] to steer the model towards factual alignment with the HWT reference. Lastly, to enhance the factual accuracy of the generated text, we implement a web-based search **Naive RAG** (Gao et al., 2024), which adds relevant context to the Content Prompts. Appendix B.1.3 provides implementation details of Naive RAG.

We evaluate these prompts using standard automatic metrics: BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) for n-gram overlap, BERTScore (Zhang et al., 2019) for semantic similarity, and QAFactEval (Fabbri et al., 2022) (F1-score) as a QA-based metric for factual consistency between HWT and MGT.[12]

Language	Technique	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	QAFactEval
English	Minimal	0.02	0.29	0.06	0.17	0.76	0.06
	Content Prompts	0.22	0.57	0.31	0.44	**0.88**	0.25
	RAG	**0.25**	**0.61**	**0.35**	**0.47**	**0.88**	**0.38**
Portuguese	Minimal	0.02	0.31	0.06	0.17	0.86	0.06
	Content Prompts	0.20	0.56	0.30	0.41	0.91	0.25
	RAG	**0.25**	**0.61**	**0.37**	**0.47**	**0.92**	**0.42**
Vietnamese	Minimal	0.04	0.67	0.26	0.32	0.85	0.06
	Content Prompts	0.28	0.78	0.52	0.54	0.91	0.27
	RAG	**0.30**	**0.79**	**0.54**	**0.55**	**0.92**	**0.36**

Table 1: Paragraph Writing prompts evaluation results.

Table 1 presents our prompting evaluation results. We find that our Naive RAG approach consistently outperforms both Minimal and Content Prompts across subtasks and languages. The low evaluation scores for Minimal prompts highlight that MGT produced in prior work is often synthetically divergent from its human-written references. While Content Prompts substantially improve performance, Naive RAG further enhances generation quality, particularly in terms of factual consistency, which is critical for encyclopaedic content.[13] Based on these findings, we adopt Naive RAG as the prompting strategy for the Paragraph Writing task in our MGT detection experiments.

4.2 Summarisation

Summarisation tasks the model with generating a lead section of comparable length to the human-written reference, based on the article's body and infobox, both of which are the main sources for lead section information (Gao et al., 2021). We frame this as a single-document, abstractive summarisation task, following Wikipedia's Manual of Style[14] and prior work on Wikipedia summarisation (Casola et al., 2021; Gao et al., 2021; Perez-Beltrachini and Lapata, 2022). Compared to *Paragraph Writing*, this *content condensation* task involves slightly less LLM contribution due to its stronger grounding in existing article content.

We use three prompting techniques from the literature on LLM-generated summaries (Goyal et al., 2022; Pu et al., 2023; Zhang et al., 2023) that align

[10] https://platform.openai.com/docs/models/gpt-4o-mini

[11] https://openai.com/index/gpt-4o-system-card/

[12] For Portuguese and Vietnamese texts, QAFactEval evaluations were performed using GPT-4 translations.

[13] https://en.wikipedia.org/wiki/Wikipedia:Verifiability

[14] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section

with this editing scenario. Each prompt contains the article content as input and conditions the output length on the target lead length. **Minimal** is a simple zero-shot baseline prompt that instructs the model to summarise the article content. **Instruction** adds a concise definition of, and instructions for compiling, a lead section to guide the model more explicitly. **Few-shot** further includes 1–3 high-quality lead–content examples, retrieved from the respective Wikipedia Featured Articles page, in addition to the Instruction prompt to enable in-context learning (Brown et al., 2020).[15] We evaluate these prompts using traditional automatic metrics for summarisation evaluation (see Section 4.1).

Language	Technique	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	QAFactEval
English	Minimal	0.06	0.37	0.13	0.26	0.79	0.45
	Instruction	0.13	0.44	0.21	0.33	0.82	**0.46**
	One-shot	**0.18**	**0.47**	**0.24**	**0.36**	**0.83**	**0.46**
	Two-shot	**0.18**	**0.47**	**0.24**	**0.36**	**0.83**	**0.46**
	Three-shot	0.16	0.46	0.23	0.35	**0.83**	**0.46**
Portuguese	Minimal	0.06	0.35	0.13	0.23	0.87	**0.48**
	Instruction	**0.11**	0.42	0.19	0.30	**0.88**	**0.48**
	One-shot	**0.11**	0.42	0.19	0.29	**0.88**	**0.48**
	Two-shot	**0.11**	**0.43**	0.19	**0.30**	**0.88**	0.47
	Three-shot	0.12	**0.43**	**0.20**	**0.30**	**0.88**	0.47
Vietnamese	Minimal	0.07	0.63	0.28	0.35	0.86	**0.45**
	Instruction	0.11	0.64	0.31	**0.38**	**0.87**	0.43
	One-shot	**0.12**	0.65	**0.32**	**0.38**	**0.87**	**0.45**
	Two-shot	**0.12**	**0.66**	**0.32**	**0.38**	**0.87**	0.44
	Three-shot	0.11	0.65	**0.32**	**0.38**	**0.87**	0.42

Table 2: Summarisation prompts evaluation results.

Table 2 presents the summarisation prompt evaluation results, showing that across languages, Instruction and Few-shot achieve higher overlap and semantic similarity scores, although Few-shot only marginally improves over Instruction. Factuality scores remain relatively stable across prompts, presumably because summarisation is a core task in aligning LLMs through reinforcement learning from human feedback (Ouyang et al., 2022). Given that increasing the number of shots does not yield further improvements, and considering the context window of smaller LLMs, we select one-shot prompting for our experiments.

4.3 Text Style Transfer

We adopt the TST task of *neutralising* revision-level NPOV violations, as introduced by Pryzant et al. (2020). In our setup, the model is instructed to revise a biased sentence or paragraph with minimal edits, aligning the output with Wikipedia's neutrality guidelines. While various TST tasks are possible on Wikipedia, focusing on NPOV violations ensures direct alignment with one of its core content policies.[16] This *content modification* task involves the least LLM contribution, as the model is conditioned to perform only minor revisions to existing text.

We test three prompting techniques for TST that are conceptually identical to those used in summarisation and align with recent work on LLM-based TST (Reif et al., 2021; Dwivedi-Yu et al., 2022; Ashkinaze et al., 2024). All prompts include the biased input text and constrain the output to be no longer than the target text. Compared to summarisation, **Minimal** instructs the model to neutralise the input; **Instruction** adds a concise definition of Wikipedia's NPOV policy; and **Few-shot** includes 1–5 randomly sampled biased–neutralised examples.

We evaluate these TST prompts along two dimensions: *semantic content preservation*, for which we report BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2019); and *style transfer accuracy*, for which we fine-tune pre-trained language models for each language and report the accuracy of binary style classification. Fine-tuning details are provided in Appendix B.3.

Language	Technique	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	ST
English	Minimal	0.35	0.68	0.52	0.66	0.92	0.90
	Instruction	0.36	0.68	0.52	0.66	0.92	**0.94**
	One-shot	0.52	0.78	0.65	0.76	**0.95**	0.91
	Two-shot	0.47	0.75	0.61	0.73	0.94	0.90
	Three-shot	0.54	0.79	0.67	0.78	**0.95**	0.89
	Four-shot	**0.56**	**0.80**	**0.69**	**0.79**	**0.95**	0.89
	Five-shot	**0.55**	**0.80**	0.68	0.78	**0.95**	0.91
Portuguese	Minimal	0.41	0.71	0.58	0.69	0.94	0.86
	Instruction	0.40	0.70	0.57	0.67	0.94	0.88
	One-shot	0.50	0.75	0.64	0.74	**0.96**	0.90
	Two-shot	0.51	0.77	0.65	0.75	**0.96**	0.89
	Three-shot	0.53	0.78	0.66	0.76	**0.96**	0.91
	Four-shot	**0.58**	**0.81**	**0.70**	**0.79**	**0.96**	**0.92**
	Five-shot	0.55	0.79	0.68	0.77	**0.96**	0.91
Vietnamese	Minimal	0.43	0.78	0.65	0.73	0.95	0.84
	Instruction	0.45	0.80	0.67	0.73	0.94	0.79
	One-shot	0.44	0.78	0.66	0.71	0.95	**0.88**
	Two-shot	0.51	0.82	0.70	0.76	0.95	0.87
	Three-shot	0.50	0.81	0.70	0.75	0.95	0.85
	Four-shot	0.51	0.82	0.70	0.76	0.95	0.85
	Five-shot	**0.55**	**0.83**	**0.73**	**0.78**	**0.96**	0.84
English Para.	Minimal	0.35	0.68	0.52	0.66	0.92	0.97
	Instruction	0.36	0.68	0.52	0.66	0.92	**0.99**
	One-shot	0.52	0.78	0.65	0.76	**0.95**	0.95
	Two-shot	0.47	0.75	0.61	0.73	0.94	0.98
	Three-shot	0.54	0.79	0.67	0.78	**0.95**	0.96
	Four-shot	**0.56**	**0.80**	**0.69**	**0.79**	**0.95**	0.95
	Five-shot	0.55	**0.80**	0.68	0.78	**0.95**	0.96

Table 3: TST prompts evaluation results.

Table 3 presents the prompt evaluation metrics for the TST task, evaluated at the sentence level for all languages, and additionally at the paragraph level for English. Across languages and levels, we find that four- and five-shot prompting consistently outperforms Minimal and Instruction

[15] https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

[16] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

prompts. While differences in semantic similarity and style transfer are marginal across prompts, we observe substantial improvements in overlap-based metrics as the number of few-shot examples increases. These improvements can be attributed to the fact that neutralisation edits in mWNC tend to be relatively minimal. For instance, in the English sentence subset, on average only 14% of words are deleted and 7% added—similar trends hold for the other subsets. As a result, the model appears to learn from the examples to apply similarly sparse edits, thereby producing outputs that match the reference text more closely in terms of n-gram overlap. Based on these findings, we adopt five-shot prompting to generate MGT in our subsequent experiments.

5 Experimental Setup

In this section, we introduce generators, detectors, benchmark construction, and evaluation metrics.

Generators We generate MGT using four multilingual models from two families: proprietary and open-weight. We select models based on their ranking at the time of writing on LM Arena,[17] an open-source platform for crowdsourced AI benchmarking. For proprietary models, we use **GPT-4o mini**[18] and **Gemini 2.0 Flash**.[19] For open-weight models, we select **Qwen2.5-7B-Instruct**[20] and **Mistral-7B-Instruct**.[21] We opt for smaller models in this category to better align with our editor-driven writing task scenarios.

Detectors We evaluate six detectors from three different families: training-based, zero-shot white-box, and zero-shot black-box methods. We consider only multilingual LLMs for all families. Specifically, we use **XLM-RoBERTa** (Conneau et al., 2020) and **mDeBERTa** (He et al., 2023) as training-based detectors, which we fine-tune with hyperparameter search; **Binoculars** (Hans et al., 2024), **LLR** (Su et al., 2023), and **FastDetectGPT (White-Box)** (Hans et al., 2024) as zero-shot white-box detectors; and **Revise-Detect** (Zhu et al., 2023a), **GECScore** (Wu et al., 2025), and **FastDetectGPT (Black-Box)** (Hans et al., 2024) as zero-shot black-box detectors. Appendix C provides an overview and implementation details of each detector.

WETBench We construct our benchmarking data by randomly sampling 2,700 HWT per task from the corresponding subsets of WikiPS and mWNC. For Paragraph Writing and Summarisation, we balance each subset by length tertiles; for TST, we evaluate at the sentence level for all languages and at the paragraph level for English only. For each task–language subset, we generate MGT using the four generators introduced above, applying the best-performing prompts from our prompt evaluation in Section 4: Naive RAG for Paragraph Writing, one-shot prompting for Summarisation, and five-shot prompting for TST. Our benchmark corpus comprises 101,940 human- and machine-written texts across tasks, languages, and generators. Appendix Table 6 presents benchmark statistics.

Evaluation Metrics Given the parallel nature of our benchmark data, our main evaluation metric is accuracy. We additionally report F1-scores, which represent the weighted harmonic mean of precision and recall.

6 Results

Table 4 presents our benchmarking results. Our main results are: (*i*) our benchmark challenges detectors, which achieve considerably lower scores than in prior work (e.g., Macko et al., 2023; Guo et al., 2023; Li et al., 2023; Wang et al., 2023, 2024a), (*ii*) supervised detectors significantly outperform zero-shot methods across all tasks and languages, and (*iii*) detection accuracy is highest for summarisation, followed by slightly lower accuracy for paragraph writing, and lowest for TST. The following presents the most relevant trends by task.

Paragraph Writing Across languages and models, training-based detectors outperform zero-shot methods by 19–30% accuracy on average. Black-box detectors are 3–6% more accurate than white-box detectors in English and Portuguese but perform slightly worse in Vietnamese. Only white-box detectors show a slight increase in accuracy when moving from high- to low-resource languages.

[17] https://blog.lmarena.ai/

[18] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[19] https://deepmind.google/technologies/gemini/flash/

[20] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

[21] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Task	Detector	English										Portuguese										Vietnamese									
		GPT-4o mini		Gemini 2.0		Qwen 2.5		Mistral		**Avg**		GPT-4o mini		Gemini 2.0		Qwen 2.5		Mistral		**Avg**		GPT-4o mini		Gemini 2.0		Qwen 2.5		Mistral		**Avg**	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Introductory Paragraph	Binoculars	0.61	0.60	0.58	0.61	0.60	0.58	0.55	0.63	**0.59**	**0.60**	0.68	0.66	0.64	0.64	0.64	0.60	0.54	0.61	**0.62**	**0.63**	0.77	0.77	0.72	0.72	0.70	0.66	0.50	0.67	**0.67**	**0.70**
	LLR	0.52	0.51	0.50	0.67	0.50	0.67	0.53	0.62	**0.51**	**0.62**	0.57	0.51	0.54	0.51	0.50	0.67	0.53	0.51	**0.53**	**0.55**	0.63	0.60	0.58	0.54	0.51	0.19	0.50	0.00	**0.56**	**0.33**
	FDGPT (WB)	0.59	0.60	0.54	0.52	0.52	0.43	0.52	0.52	**0.54**	**0.51**	0.68	0.66	0.63	0.63	0.56	0.53	0.52	0.57	**0.60**	**0.60**	0.76	0.77	0.70	0.69	0.59	0.53	0.50	0.00	**0.64**	**0.50**
	Avg. White-box	0.57	0.57	0.54	0.60	0.54	0.56	0.54	0.59	**0.55**	**0.58**	0.64	0.61	0.60	0.59	0.56	0.60	0.53	0.56	**0.58**	**0.59**	0.72	0.71	0.67	0.65	0.60	0.46	0.50	0.22	**0.62**	**0.51**
	Revise	0.53	0.41	0.53	0.50	0.52	0.55	0.52	0.62	**0.52**	**0.52**	0.55	0.58	0.56	0.50	0.54	0.53	0.52	0.59	**0.54**	**0.55**	0.53	0.50	0.54	0.56	0.54	0.59	0.50	0.00	**0.53**	**0.41**
	GECScore	0.82	0.82	0.77	0.75	0.77	0.75	0.72	0.73	**0.77**	**0.76**	0.79	0.79	0.80	0.80	0.65	0.66	0.54	0.66	**0.69**	**0.73**	0.72	0.70	0.70	0.70	0.58	0.47	0.50	0.67	**0.62**	**0.63**
	FDGPT (BB)	0.58	0.61	0.53	0.49	0.52	0.45	0.54	0.50	**0.55**	**0.51**	0.69	0.66	0.62	0.60	0.56	0.44	0.54	0.42	**0.60**	**0.53**	0.74	0.74	0.68	0.70	0.59	0.59	0.50	0.00	**0.63**	**0.51**
	Avg. Black-box	0.64	0.61	0.61	0.58	0.60	0.58	0.59	0.62	**0.61**	**0.60**	0.68	0.68	0.66	0.63	0.58	0.54	0.53	0.56	**0.61**	**0.60**	0.66	0.65	0.64	0.65	0.57	0.55	0.50	0.22	**0.59**	**0.52**
	xlm-RoBERTa	0.81	0.81	0.84	0.84	0.85	0.85	0.83	0.83	**0.83**	**0.83**	0.76	0.75	0.80	0.79	0.83	0.82	0.76	0.74	**0.79**	**0.78**	0.75	0.72	0.82	0.81	0.89	0.89	0.98	0.98	**0.86**	**0.85**
	mDeBERTa	0.86	0.86	0.84	0.83	0.89	0.88	0.90	0.89	**0.87**	**0.87**	0.76	0.74	0.80	0.79	0.84	0.83	0.85	0.85	**0.81**	**0.80**	0.77	0.76	0.82	0.81	0.87	0.87	0.98	0.98	**0.86**	**0.86**
	Avg. Supervised	0.84	0.83	0.84	0.84	0.87	0.87	0.86	0.86	**0.85**	**0.85**	0.76	0.75	0.80	0.79	0.83	0.83	0.80	0.79	**0.80**	**0.79**	0.76	0.74	0.82	0.81	0.88	0.88	0.98	0.98	**0.86**	**0.85**
Summarisation	Binoculars	0.60	0.60	0.62	0.58	0.62	0.62	0.62	0.69	**0.61**	**0.62**	0.72	0.73	0.70	0.72	0.71	0.72	0.62	0.66	**0.69**	**0.71**	0.72	0.72	0.70	0.71	0.72	0.73	0.50	0.67	**0.66**	**0.71**
	LLR	0.52	0.65	0.52	0.64	0.54	0.66	0.61	0.68	**0.55**	**0.66**	0.58	0.57	0.56	0.54	0.54	0.65	0.55	0.67	**0.56**	**0.61**	0.58	0.47	0.55	0.47	0.54	0.65	0.51	0.67	**0.55**	**0.56**
	FDGPT (WB)	0.60	0.59	0.60	0.59	0.55	0.51	0.61	0.60	**0.59**	**0.57**	0.72	0.70	0.69	0.69	0.65	0.63	0.56	0.58	**0.65**	**0.65**	0.73	0.72	0.71	0.71	0.64	0.62	0.50	0.00	**0.64**	**0.51**
	Avg. White-box	0.57	0.61	0.58	0.61	0.57	0.60	0.61	0.66	**0.58**	**0.62**	0.67	0.67	0.65	0.65	0.63	0.67	0.58	0.64	**0.63**	**0.66**	0.68	0.64	0.66	0.63	0.64	0.67	0.50	0.45	**0.62**	**0.59**
	Revise	0.53	0.61	0.53	0.58	0.53	0.62	0.53	0.61	**0.53**	**0.60**	0.54	0.51	0.53	0.57	0.53	0.55	0.51	0.63	**0.53**	**0.56**	0.53	0.57	0.54	0.56	0.54	0.56	0.50	0.66	**0.53**	**0.59**
	GECScore	0.80	0.81	0.71	0.70	0.75	0.75	0.74	0.76	**0.75**	**0.75**	0.78	0.79	0.73	0.72	0.68	0.70	0.52	0.64	**0.68**	**0.71**	0.71	0.70	0.67	0.66	0.63	0.64	0.50	0.67	**0.63**	**0.67**
	FDGPT (BB)	0.59	0.58	0.59	0.63	0.56	0.60	0.63	0.62	**0.59**	**0.61**	0.71	0.69	0.67	0.70	0.65	0.66	0.57	0.57	**0.65**	**0.65**	0.70	0.68	0.68	0.66	0.64	0.61	0.50	0.01	**0.63**	**0.49**
	Avg. Black-box	0.64	0.66	0.61	0.63	0.61	0.66	0.63	0.66	**0.62**	**0.65**	0.67	0.66	0.64	0.66	0.62	0.65	0.53	0.61	**0.62**	**0.64**	0.65	0.65	0.63	0.63	0.60	0.60	0.50	0.45	**0.60**	**0.58**
	xlm-RoBERTa	0.92	0.92	0.86	0.85	0.95	0.95	0.90	0.90	**0.91**	**0.90**	0.91	0.91	0.84	0.84	0.91	0.91	0.94	0.94	**0.90**	**0.90**	0.86	0.86	0.78	0.77	0.94	0.93	0.94	0.94	**0.88**	**0.87**
	mDeBERTa	0.91	0.91	0.83	0.83	0.93	0.93	0.91	0.90	**0.89**	**0.89**	0.89	0.89	0.86	0.85	0.94	0.93	0.94	0.94	**0.91**	**0.90**	0.84	0.84	0.77	0.76	0.90	0.90	0.96	0.96	**0.87**	**0.87**
	Avg. Supervised	0.91	0.91	0.84	0.84	0.94	0.94	0.90	0.90	**0.90**	**0.90**	0.90	0.90	0.85	0.85	0.92	0.92	0.94	0.94	**0.90**	**0.90**	0.85	0.85	0.78	0.76	0.92	0.92	0.95	0.95	**0.87**	**0.87**
Text Style Transfer	Binoculars	0.53	0.46	0.52	0.33	0.50	0.05	0.53	0.54	**0.52**	**0.34**	0.56	0.54	0.55	0.51	0.53	0.49	0.52	0.48	**0.54**	**0.51**	0.59	0.55	0.57	0.52	0.53	0.47	0.50	0.00	**0.55**	**0.39**
	LLR	0.50	0.02	0.50	0.05	0.50	0.02	0.50	0.04	**0.50**	**0.03**	0.51	0.25	0.50	0.03	0.50	0.02	0.50	0.32	**0.50**	**0.15**	0.52	0.29	0.51	0.16	0.51	0.26	0.50	0.67	**0.51**	**0.34**
	FDGPT (WB)	0.54	0.55	0.53	0.54	0.50	0.67	0.53	0.54	**0.52**	**0.57**	0.56	0.53	0.54	0.51	0.51	0.44	0.51	0.49	**0.53**	**0.49**	0.58	0.57	0.55	0.55	0.53	0.52	0.50	0.00	**0.54**	**0.41**
	Avg. White-box	0.52	0.34	0.52	0.31	0.50	0.24	0.52	0.37	**0.52**	**0.32**	0.54	0.44	0.53	0.35	0.51	0.32	0.51	0.43	**0.52**	**0.38**	0.56	0.47	0.54	0.41	0.52	0.42	0.50	0.22	**0.53**	**0.38**
	Revise	0.53	0.63	0.52	0.60	0.52	0.65	0.52	0.62	**0.52**	**0.62**	0.53	0.56	0.52	0.59	0.52	0.53	0.51	0.59	**0.52**	**0.56**	0.53	0.59	0.54	0.49	0.51	0.66	0.51	0.66	**0.52**	**0.60**
	GECScore	0.65	0.64	0.62	0.59	0.64	0.62	0.62	0.60	**0.63**	**0.61**	0.66	0.63	0.62	0.59	0.61	0.60	0.55	0.54	**0.61**	**0.59**	0.63	0.57	0.57	0.47	0.56	0.48	0.50	0.00	**0.57**	**0.38**
	FDGPT (BB)	0.53	0.55	0.51	0.49	0.50	0.01	0.53	0.52	**0.52**	**0.39**	0.55	0.57	0.53	0.52	0.51	0.22	0.52	0.32	**0.53**	**0.41**	0.57	0.53	0.53	0.46	0.52	0.51	0.50	0.00	**0.53**	**0.37**
	Avg. Black-box	0.57	0.61	0.55	0.56	0.55	0.42	0.56	0.58	**0.56**	**0.54**	0.58	0.59	0.56	0.57	0.55	0.45	0.52	0.48	**0.55**	**0.52**	0.58	0.56	0.55	0.47	0.53	0.55	0.50	0.22	**0.54**	**0.45**
	xlm-RoBERTa	0.64	0.59	0.59	0.57	0.59	0.58	0.64	0.61	**0.61**	**0.59**	0.63	0.60	0.66	0.65	0.58	0.58	0.68	0.67	**0.64**	**0.63**	0.63	0.62	0.68	0.68	0.54	0.52	0.52	0.45	**0.59**	**0.57**
	mDeBERTa	0.62	0.58	0.60	0.60	0.58	0.58	0.62	0.62	**0.61**	**0.59**	0.64	0.58	0.68	0.68	0.66	0.66	0.69	0.69	**0.67**	**0.65**	0.62	0.58	0.69	0.69	0.61	0.60	0.64	0.64	**0.64**	**0.63**
	Avg. Supervised	0.63	0.59	0.60	0.59	0.59	0.58	0.63	0.62	**0.61**	**0.59**	0.63	0.59	0.67	0.67	0.62	0.62	0.68	0.68	**0.65**	**0.64**	0.63	0.60	0.69	0.68	0.57	0.56	0.58	0.54	**0.62**	**0.60**

Table 4: Detection accuracy (ACC) and F1-scores (F1) across tasks, languages, and models. Gray highlights average performances across detector families by generator (rows) and across generators by detector (bold columns).

Within detector families, we do not find any substantial differences between the two training-based models. Among white-box detectors, Binoculars achieves up to 11% higher accuracy, with performance on low-resource languages approaching that of training-based methods. For black-box detectors, GECScore exhibits up to 25% higher accuracy compared to other models in its category.

Considering generators, training-based detectors achieve on average 2–7% higher accuracy on smaller-sized generators. This pattern is reversed for zero-shot detectors, where accuracy is higher for larger models, with the gap widening in lower-resource languages. These results suggest that when generating paragraphs from scratch, smaller models leave more detectable semantic and syntactic traces for training-based detectors. In contrast, the internals and token-level patterns of larger models seem to exhibit stronger signals than those of smaller models for zero-shot methods.

We observe substantial anomalies in Mistral's output for Vietnamese. The model often fails to follow prompts, producing unclear outputs that provide simple cues for training-based detectors but appear to confuse zero-shot methods. We provide an analysis of Mistral's generation issues in Appendix D.1.

Summarisation Among all tasks, supervised detectors perform best on summarisation, achieving an average accuracy of 89% across languages and generators. An exception is Gemini 2.0, for which detection accuracies are on average 6–17% lower, suggesting that its summaries may more closely resemble human-written references. While most LLMs are trained on summarisation tasks, enabling strong zero-shot performance (Ouyang et al., 2022), Wikipedia lead sections follow a distinctive style and formatting that seem to provide strong cues for training-based detectors.

Compared to Paragraph Writing, average detection accuracies for zero-shot models are slightly higher. White-box detectors achieve 4% higher accuracy on English summaries compared to black-box models, while performance is similar for Portuguese and Vietnamese. As in Paragraph Writing, Binoculars achieves the highest average accuracy (65%) among black-box detectors across languages and models, while GECScore performs best among white-box methods (68%).

In contrast to Paragraph Writing, zero-shot metrics show little variation across generators for English. However, for Portuguese and Vietnamese, a similar pattern emerges: summaries generated by larger models are slightly easier to detect. This effect is less pronounced than in Paragraph Writing, with an average accuracy difference of around 6%.

We attribute the similar trends between Paragraph Writing and Summarisation to the nature of both tasks: each involves generating text from scratch, conditioned either on retrieved context or article content. We observe the same issues as before for Mistral's Vietnamese summaries.

TST For sentence-level TST, we observe the lowest accuracy scores across all tasks and detector

Detector	TST English Paragraphs									
	GPT-4o mini		Gemini 2.0		Qwen 2.5		Mistral		**Avg**	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Binoculars	0.58	0.53	0.55	0.47	0.52	0.39	0.57	0.51	**0.56**	**0.48**
LLR	0.52	0.25	0.51	0.22	0.50	0.03	0.53	0.40	**0.51**	**0.22**
FDGPT (WB)	0.60	0.63	0.56	0.60	0.52	0.60	0.58	0.61	**0.56**	**0.61**
Avg (White-box)	0.57	0.47	0.54	0.43	0.52	0.34	0.56	0.51	**0.55**	**0.44**
Revise	0.53	0.62	0.52	0.56	0.53	0.42	0.52	0.55	**0.53**	**0.54**
GECScore	0.83	0.82	0.64	0.67	0.73	0.69	0.67	0.69	**0.72**	**0.72**
FDGPT (BB)	0.59	0.62	0.54	0.55	0.51	0.63	0.58	0.54	**0.56**	**0.59**
Avg (Black-box)	0.65	0.69	0.57	0.59	0.59	0.58	0.59	0.59	**0.60**	**0.61**
xlm-RoBERTa	0.78	0.77	0.78	0.77	0.78	0.77	0.71	0.71	**0.76**	**0.76**
mDeBERTa	0.83	0.83	0.77	0.76	0.81	0.81	0.67	0.64	**0.77**	**0.76**
Avg (Supervised)	0.81	0.80	0.78	0.77	0.80	0.79	0.69	0.67	**0.77**	**0.76**

Table 5: Detection accuracy (ACC) and F1-scores (F1) for TST English paragraphs. Gray highlights average performances across detector families by generator (rows) and across generators by detector (bold columns).

families. While zero-shot detectors average between 52–56% across languages and generators, training-based methods achieve only slightly higher scores, ranging from 61–65%. A notable exception is GECScore, which outperforms other zero-shot methods by up to 12%.

We attribute part of the reduced performance to the sentence-level setting. Comparing the English sentence-level results in Table 4 to the paragraph-level results in Table 5, we observe accuracy gains of up to 18%, depending on the model. However, these improvements mostly apply to white-box and training-based models.

Compared to Paragraph Writing and Summarisation, TST involves only minimal modifications to human-written text. While detection scores on English paragraphs are slightly lower than for full generation from scratch, they remain substantially higher than for sentence-level TST. This suggests that training-based detectors can identify similarly strong MGT signals in paragraph-level text, regardless of whether the content is generated from scratch or modified at the token level.

7 Conclusion

We present **WETBench**, a multilingual, multi-generator benchmark for detecting MGT in task-specific Wikipedia editing scenarios. We build the benchmark from two new large-scale, multilingual Wikipedia text corpora—**WikiPS** and **mWNC**—which support a range of tasks relevant to the Wikipedia and AI communities. Based on these data, we define three representative tasks, evaluate multiple prompting strategies, generate MGT from diverse LLMs using the best-performing prompts, and benchmark detectors.

Our benchmark reveals that detectors from diverse families underperform on our data, with substantial variation across languages, models, and tasks. Training-based detectors consistently outperform zero-shot methods but achieve only moderate detection accuracy. These results indicate that existing detectors struggle to generalise beyond generic setups, highlighting uncertainty around their reliability and effectiveness in real-world, editor-driven MGT scenarios on Wikipedia.

In future work, we plan to extend the benchmark with additional tasks, generators, and languages. We also aim to investigate the generalisability of our findings to open-ended generation tasks and other domains.

Limitations

Editing Task Selection We identify three common editing tasks, based on Ford et al. (2023), that vary in editing intensity. However, many other relevant editing tasks exist, reflecting different forms of content transformation. In particular, *text translation* is a critical use case across many language editions of Wikipedia, as it helps bridge content gaps. Given the increasing capabilities of LLMs in translation (Jiao et al., 2023; Zhu et al., 2023b; Yan et al., 2024), and the associated risks (see Section 1), detecting machine-generated translations is an important and underexplored task. Similarly, there are alternative approaches to TST, such as grammar and spelling correction, which are highly relevant, especially for non-native Wikipedia editors.

Real-World Relevance of Editing Tasks Our task selection is grounded in the study by Ford et al. (2023), which explores how Wikipedia editors perceive opportunities for AI-assisted writing. However, we acknowledge that our benchmark does not fully capture how MGT actually arises in real-world Wikipedia usage. While our tasks are motivated by plausible scenarios, we lack empirical evidence that editors systematically use LLMs in the ways we design them. Nonetheless, the findings of Ford et al. (2023) provide the most systematic basis for aligning our benchmark with real-world editorial contexts.

NPOV Detection To identify the most effective prompting technique for TST, we train four style classifiers per language-level setting (see Appendix Table 9). However, our classifiers for Vietnamese and English at the paragraph level achieve accuracy

only slightly above random chance, which might compromise the prompt evaluation in Section 4.3. Despite extensive fine-tuning across model types, data, and hyperparameters, performance remains limited. For both subsets, we report the most conservative results to ensure that, even if classifier performance is poor, the precision of NPOV-related revisions is maximised (see Appendix B.3 for details). We acknowledge that NPOV detection on Vietnamese and English paragraph-level data is intrinsically challenging.

Text Length When comparing detection results between sentence- and paragraph-level TST, we find that text length significantly affects performance. While we stratify samples by tertiles to control for length, we do not further analyse detection performance based on length, instead reporting average metrics. Given its impact, we plan to investigate text-length heterogeneity in future work.

Generalisability Although we aim to cover a broad range of detectors, generators, and languages, our conclusions are limited to the evaluated settings. Due to the rapid pace of AI research, our configurations may quickly become outdated. For example, through advances in LLMs or MGT detectors. To support ongoing progress, we open-source our data and benchmark and plan to maintain the repository to ensure its continued relevance.

Ethics Statement

Our work uses publicly available content from Wikipedia, licensed under CC BY-SA. We include no private or sensitive information, and our experiments pose no risk to Wikipedia editors or the Wikipedias under study. Sensitive data about individual contributors are neither identifiable nor exposed in any way.

We obtain machine-generated data using four LLMs under their respective licences:

- GPT-4-mini: No specific license. OpenAI welcomes research publications.[22]
- Gemini 2.0: Apache 2.0[23]
- QWen 2.0: Apache 2.0[24]
- Mistral: Apache 2.0[25]

This study addresses limitations in prior evaluations of SOTA MGT detectors by systematically assessing their performance in realistic editorial contexts. Our goal is to provide more accurate and practical insights into the feasibility and utility of MGT detection in collaborative knowledge environments such as Wikipedia. We emphasise that our experiments aim to inform the potential role of MGT detectors as automated metrics or as tools to assist users in identifying machine-generated content.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [grant number Y009800/1], through funding from Responsible AI UK (KP0011), as part of the Participatory Harm Auditing Workbenches and Methodologies (PHAWM) project. Additional support was provided by UK Research and Innovation [grant number EP/S023356/1] through the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence (www.safeandtrustedai.org).

References

Joshua Ashkinaze, Ruijia Guan, Laura Kurek, Eytan Adar, Ceren Budak, and Eric Gilbert. 2024. Seeing like an ai: How llms apply (and misapply) wikipedia neutrality norms. *Preprint*, arXiv:2407.04183.

Nishant Balepur, Jie Huang, and Kevin Chang. 2023. Expository text generation: Imitate, retrieve, paraphrase. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11896–11919, Singapore. Association for Computational Linguistics.

Siddhartha Banerjee and Prasenjit Mitra. 2015. WikiKreator: Improving Wikipedia stubs automatically. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 867–877, Beijing, China. Association for Computational Linguistics.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

[22] https://openai.com/policies/sharing-publication-policy/

[23] https://github.com/google-gemini

[24] https://github.com/QwenLM/Qwen2.5

[25] https://mistral.ai/news/announcing-mistral-7b

Creston Brooks, Samuel Eggert, and Denis Peskoff. 2024. The rise of ai-generated content in wikipedia. *Preprint*, arXiv:2410.08044.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Silvia Casola, Alberto Lavelli, and 1 others. 2021. Wits: Wikipedia for italian text summarization. In *CLiC-it*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *Preprint*, arXiv:1911.02116.

Selena Deckelmann. 2023. Wikipedia's value in the age of generative ai.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. Editeval: An instruction-based benchmark for text improvements. *arXiv preprint arXiv:2209.13331*.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization. *Preprint*, arXiv:2112.08542.

Angela Fan and Claire Gardent. 2022. Generating biographies on Wikipedia: The impact of gender bias on the retrieval-based generation of women biographies. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8561–8576, Dublin, Ireland. Association for Computational Linguistics.

Heather Ford, Michael Davis, and Timothy Koskie. 2023. Implications of chatgpt for knowledge integrity on wikipedia. Accessed: 2025-04-06.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Preprint*, arXiv:2309.00770.

Shen Gao, Xiuying Chen, Chang Liu, Dongyan Zhao, and Rui Yan. 2021. Biogen: Generating biography summary under table guidance on wikipedia. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4752–4757.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Demian Gholipour Ghalandari, Chris Hokamp, Nghia The Pham, John Glover, and Georgiana Ifrim. 2020. A large-scale multi-document summarization dataset from the wikipedia current events portal. *Preprint*, arXiv:2005.10070.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2022. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Hiroaki Hayashi, Prashant Budania, Peng Wang, Chris Ackerson, Raj Neervannan, and Graham Neubig. 2021. Wikiasp: A dataset for multi-domain aspect-based summarization. *Transactions of the Association for Computational Linguistics*, 9:211–225.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and

open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Siming Huang, Yuliang Xu, Mingmeng Geng, Yao Wan, and Dongping Chen. 2025b. Wikipedia in the era of llms: Evolution and risks. *arXiv preprint arXiv:2503.02879*.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine. *arXiv preprint arXiv:2301.08745*.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen McKeown. 2020. WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online. Association for Computational Linguistics.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Mage: Machine-generated text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, toxicity. *Preprint*, arXiv:2305.13169.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Samuel Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and 1 others. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark. *arXiv preprint arXiv:2310.13606*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature (2023). *arXiv preprint arXiv:2301.11305*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, and 1 others. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Laura Perez-Beltrachini and Mirella Lapata. 2022. Models and datasets for cross-lingual summarisation. *arXiv preprint arXiv:2202.09583*.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Hongjing Qian, Yutao Zhu, Zhicheng Dou, Haoqi Gu, Xinyu Zhang, Zheng Liu, Ruofei Lai, Zhao Cao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Webbrain: Learning to generate factually correct articles for queries by grounding on large web corpus. *Preprint*, arXiv:2304.04358.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Preprint*, arXiv:1606.05250.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Marija Sakota, Maxime Peyrard, and Robert West. 2023. Descartes: Generating short descriptions of wikipedia articles. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1446–1456, New York, NY, USA. Association for Computing Machinery.

K. Salas-Jimenez, Francisco Fernando Lopez-Ponce, Sergio-Luis Ojeda-Trueba, and Gemma Bel-Enguix. 2024. WikiBias as an extrapolation corpus for

bias detection. In *Proceedings of the First Workshop on Advancing Natural Language Processing for Wikipedia*, pages 46–52, Miami, Florida, USA. Association for Computational Linguistics.

Christina Sauper and Regina Barzilay. 2009. Automatically generating Wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore. Association for Computational Linguistics.

Yijia Shao, Yucheng Jiang, Theodore A. Kanell, Peter Xu, Omar Khattab, and Monica S. Lam. 2024. Assisting in writing wikipedia-like articles from scratch with large language models. *Preprint*, arXiv:2402.14207.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

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

Zhen Tao, Zhiyu Li, Dinghao Xi, and Wei Xu. 2024. Cudrt: Benchmarking the detection of human vs. large language models generated texts. *arXiv preprint arXiv:2406.09056*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, and 1 others. 2024a. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. *arXiv preprint arXiv:2402.11175*.

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. 2024b. 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)*, Bangkok, Thailand. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, and Min Zhang. 2025. Who wrote this? the key to zero-shot llm-generated text detection is gecscore. *Preprint*, arXiv:2405.04286.

Jianhao Yan, Pingchuan Yan, Yulong Chen, Judy Li, Xianchao Zhu, and Yue Zhang. 2024. Gpt-4 vs. human translators: A comprehensive evaluation of translation quality across languages, domains, and expertise levels. *arXiv preprint arXiv:2407.03658*.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of llms-generated content. *arXiv preprint arXiv:2310.15654*.

Jiebin Zhang, Eugene J. Yu, Qinyu Chen, Chenhao Xiong, Dawei Zhu, Han Qian, Mingbo Song, Weimin Xiong, Xiaoguang Li, Qun Liu, and Sujian Li. 2025. WIKIGENBENCH:exploring full-length Wikipedia generation under real-world scenario. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5191–5210, Abu Dhabi, UAE. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2023. Benchmarking large language models for news summarization. *Preprint*, arXiv:2301.13848.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Jingyi Zheng, Junfeng Wang, Zhen Sun, Wenhan Dong, Yule Liu, and Xinlei He. 2025. Th-bench: Evaluating evading attacks via humanizing ai text on machine-generated text detectors. *arXiv preprint arXiv:2503.08708*.

Yang Zhong, Jingfeng Yang, Wei Xu, and Diyi Yang. 2021. WIKIBIAS: Detecting multi-span subjective biases in language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1799–1814, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023a. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023b. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

A Data Construction

We download the meta stub history WikiDumps[26] for all three languages, which serve as the foundational datasets for both **WikiPS** and the **mWNC**. For both datasets, we consider only the most recent instances—revisions for mWNC and article versions for WikiPS—that occurred prior to the public release of ChatGPT on 30 November 2022. This filtering step ensures that our data is not contaminated by MGT.

A.1 WikiPS

We begin by retrieving the latest revision IDs for all *articles* (excluding discussion pages and other non-content pages) in each target language. We then randomly sample and crawl these articles by querying the MediaWiki Action API[27] until we collect 100,000 non-stub Wikipedia articles in HTML format per language. Rather than concentrating on a set of topics, we rely on a large enough random sample to provide a representative snapshot of each Wikipedia. We also rely on HTML representations, as parsing raw MediaWiki markup often leads to errors and occasional information loss (e.g., incomplete internal links).

We filter out articles lacking essential structural elements, such as a title, lead section, content sections, or references, as well as list-based articles. From the remaining articles, we use `BeautifulSoup` to pre-process, clean, and parse the HTML and extract the following components: the lead section, infobox (if available), paragraphs with their section headers (excluding sections such as "See also", "External links", etc.), and reference lists. This process yields article-level corpora of 67,267 articles in English, 56,538 in Portuguese, and 60,884 in Vietnamese.

Paragraphs To construct our paragraph-level dataset, we randomly sample 20,000 articles per language. We then define a paragraph as a block of text containing at least three sentences and a minimum of 20 characters. For each paragraph, we collect metadata including its position within the article and any associated external references. We further refine the dataset by removing paragraphs without any references and those whose token counts fall outside two standard deviations from the mean token count of the corpus. Based on the filtered corpus, we compute tertiles for each paragraph and assign each to its corresponding range (EN $(83.0, 120.0)$; PT $(88.0, 128.0)$; VI $(108.0, 160.0)$). For the Paragraph Writing task, we only consider the first paragraph following a section or subsection. The resulting raw paragraph-level corpora consist of 96,860 paragraphs in English, 72,965 in Portuguese, and 98,315 in Vietnamese.

Summaries To construct our summarisation dataset, we extract the lead section, infobox, and article body from the processed text corpora for each article. For the English and Portuguese corpora, we exclude lead sections with fewer than 10 tokens or with token lengths exceeding two standard deviations above the token mean. Similarly, we discard article bodies with fewer than 100 tokens or more than two standard deviations above the mean token count. For the Vietnamese corpus, whose article bodies are considerably longer (see Table 7), we set an upper limit of either 2,900 tokens or two standard deviations above the mean to mitigate context length constraints during model processing. As we treat each component as text input, we apply minimal markdown-like formatting to both the infobox and article body, such as rendering headers in bold. The resulting summarisation corpora consist of 53,203 lead–article pairs in English, 36,075 in Portuguese, and 45,500 in Vietnamese.

Table 7 compares our raw summarisation datasets to three commonly used benchmarks from different domains: WikiLingua (Ladhak et al., 2020) for Wikimedia content, CNN/DM (Nallapati et al., 2016) for news, and arXiv (Cohan et al., 2018) for academic writing.

On average, our summaries are considerably longer than those in WikiLingua and CNN/DM, but shorter than arXiv abstracts. The average body length in our datasets is comparable to CNN/DM but significantly shorter than arXiv. Despite this, our datasets exhibit higher ROUGE-1 and ROUGE-2 scores, indicating improved content overlap. We also observe lower compression rates (Grusky et al., 2018), meaning our summaries are proportionally longer relative to article bodies. Furthermore, our datasets show consistently higher percentages of novel unigrams, bigrams, and trigrams, suggesting a greater degree of abstractiveness.

To address the concern that a higher proportion of novel tokens may signal information asymmetry between the lead and the article body, we com-

[26] https://dumps.wikimedia.org/

[27] https://www.mediawiki.org/wiki/API:Main_page

Corpus	Subset	Level	Language	Corpus N	Processed Corpus N	Eval N	Experiment N	MGT N
extendWNC	Text Style Transfer	Sentences	EN	2,333,143	286,626	270	2,700	10,800
			PT	31,506	7,877	270	2700	10,800
			VI	13,800	1,185	270	1185	4,740
		Paragraphs	EN	4,671	4,671	270	2700	10,800
WikiPS	Paragraph Writing		EN	96,860	96,860	270	2700	10,800
			PT	72,965	72,965	270	2700	10,800
			VI	98,315	98,315	270	2700	10,800
	Summarisation		EN	67,267	53,203	270	2700	10,800
			PT	56,538	36,075	270	2700	10,800
			VI	60,884	45,500	270	2700	10,800
Total						**2,700**	**25,485**	**101,940**

Table 6: WETBench Dataset Statistics. Corpus N denotes the raw number of observations; Processed N denotes the number of observations after processing; Experiment N denotes the number of human-written texts; and MGT N denotes the total number of machine-generated texts.

Metric/Corpus	WikiLingua	CNN/DM	arXiv	WikiSums EN	WikiSums PT	WikiSums VI
Size	142,346	311,971	215,913	67,267	56,538	60,884
Summary Length	32 (19)	51 (21)	272 (572)	83 (78)	87 (95)	135 (148)
Body Length	379 (224)	690 (337)	6029 (4570)	667 (1027)	587 (1121)	940 (1800)
Infobox Length	-	-	-	61 (60)	62 (40)	95 (78)
ROUGE-1	0.13	0.14	0.07	0.17	0.18	0.30
ROUGE-2	0.05	0.08	0.04	0.06	0.06	0.16
Compression Rate	14.12	14.66	39.78	10.30	7.80	8.56
Novel Unigram %	0.38	0.20	0.15	0.53	0.62	0.49
Novel Bigram %	0.78	0.60	0.45	0.83	0.88	0.80
Novel Trigram %	0.93	0.77	0.69	0.93	0.95	0.91
Entity Sample Size	20,000	20,000	20,000	20,000	20,000	20,000
Entity F1-Score	0.06	0.21	0.04	0.14	0.17	0.13

Table 7: Summarisation Corpora Comparison. Numbers in parentheses report standard deviation.

pute entity overlap F1-scores on a 20,000-example subset of each dataset. Our results show higher entity F1-scores compared to WikiLingua, CNN/DM, and arXiv, indicating that our datasets maintain a comparable or better level of factual consistency.

Among the Wikipedias, the Vietnamese edition features leads, infoboxes, and article bodies that are approximately 30% longer than their English and Portuguese counterparts. Despite higher ROUGE scores, the comparable share of novel n-grams in Vietnamese indicates a slightly lower level of abstractiveness relative to the other language versions.

A.2 mWNC

We largely follow the procedure of Pryzant et al. (2020), with modifications to accommodate larger multilingual datasets. From each Wikidump, we extract all NPOV-related revisions made prior to the release of ChatGPT. We expand the set of NPOV-related keywords (e.g., NPOV, POV, neutral, etc.) for each Wikipedia edition based on its respective NPOV policy page.[28] This yields 2,333,143 relevant revisions for English, 31,506 for Portuguese, and 13,800 for Vietnamese.

We retrieve the corresponding diffs[29] using the MediaWiki API,[30] which we extensively clean and pre-process. To match pre- and post-neutralisation sentence pairs within each edit chunk, we first discard all unedited sentences and then apply pairwise BLEU scoring to identify the highest-scoring sentence pairs. For details on chunk and sentence filtering, we refer to Pryzant et al. (2020).

Our main modifications include: (*1*) retaining reverts, and (*2*) for Vietnamese only, relaxing the Levenshtein distance threshold to <3 and allowing up to two edit chunk pairs and multiple sentence-level matches. This adjustment addresses the comparatively low number of NPOV-related edits in Vietnamese, which would otherwise yield only a few hundred usable instances.

These modifications result in 286,626 sentence pairs for English, 7,877 for Portuguese, and 1,185 for Vietnamese. While we could further increase N for Vietnamese by loosening the filtering criteria, we find that this introduces noise and does not improve the performance of the downstream style classifier. We therefore prioritise a smaller, higher-precision dataset (see also Appendix B.3).

Due to the stark disparity in data size, we obtain paragraph-level data only for English. For this, we construct a dataset that, like the Vietnamese setup, allows multiple edit chunk and sentence-level matches. We define a paragraph-level pair as one in which at least one addition or deletion occurs in each of three adjacent sentences. This yields a dataset of 4,671 paragraph pairs.

B Task Design Details

For brevity, we present prompts in English only.

[28] English: Neutral point of view; Portuguese: Princípio da imparcialidade; Vietnamese: Thái đ trung lp

[29] https://en.wikipedia.org/wiki/Help:Diff

3027

B.1 Paragraph Writing

B.1.1 Paragraph Writing Prompts

Minimal

```
Please write the first paragraph for the section
     "{section_title}" in the Wikipedia article
    "{page_title}" using no more than {n_words}
    words. Only return the paragraph.
```

Content Prompts

```
Please write the first paragraph for the section
     "{section_title}" in the Wikipedia article
    "{page_title}".

Address the following key points in your
    response:
{content_prompts}

Use no more than {n_words} words. Only return
    the paragraph.
```

RAG

```
Use the following context to ensure factual
    accuracy when writing:
{context}

--

Please write the first paragraph for the section
     "{section_title}" in the Wikipedia article
    "{page_title}".

Address the following key points in your
    response:
{content_prompts}

Use the context above to inform your response,
    in addition to any relevant knowledge you
    have. Use no more than {n_words} words. Only
     return the paragraph in {language}.
```

B.1.2 Content Prompts

We model editors' LLM-assisted content generation through Content Prompts. For instance, an editor aiming to expand a Wikipedia article might prompt a model to generate a paragraph in response to factual questions about a specific topic (e.g., "What are London's most notable modern buildings?" or "What is London's tallest skyscraper?"), within a given section (e.g., Architecture). For each human-written paragraph in our dataset, we prompt GPT-4o (OpenAI et al., 2024) to generate a minimum of five content prompts for low-tertile paragraphs, and eight for medium- and high-tertile paragraphs. Although this method does not exhaustively cover all factual content from the HWT, it substantially improves the alignment of factual information between HWT and MGT.

B.1.3 Naive RAG

We implement a web-based Naive RAG setup to reflect an editing scenario in which an editor, in addition to providing task instructions and content prompts, also supplies relevant context to minimise factual inaccuracies. Our RAG pipeline follows the indexing, retrieval, and generation modules of the Naive variant (Gao et al., 2024), with two key modifications: we prepend the pipeline with Content Prompts and Web Search modules.

Content Prompts and Web Search For each paragraph, we generate diverse content prompts as described above. We use each content prompt to query the Google Custom Search API,[31] retrieving the top 10 most relevant URLs. From the search results, we exclude the original Wikipedia page (if applicable) as well as any unreliable sources (Shao et al., 2024).

Indexing We download the raw HTML of each scrappable web page and apply a series of preprocessing and cleaning steps. We then split each page into chunks using LangChain's RecursiveCharacterTextSplitter.[32] We compute BGE-M3[33] embeddings for each chunk and store them in a vector database.

Retrieval and Generation We treat each content prompt as a query, compute its embedding, and retrieve the two most similar chunks from the vector database based on cosine similarity. We append these retrieved chunks to the content prompt as context to guide the model's generation.

B.2 Summarisation

B.2.1 Prompts

Minimal

```
Your task is to summarize the below article with
     no more than {n_toks_trgt} words. Article:

"""{src}"""
```

Instruction/Few-Shot

```
Your task is to summarize an article to create a
     Wikipedia lead section.
- In Wikipedia, the lead section is an
    introduction to an article and a summary of
    its most important contents.
```

[31] https://developers.google.com/custom-search/v1/overview

[32] LangChain RecursiveCharacterTextSplitter documentation

[33] https://huggingface.co/BAAI/bge-m3

```
- Apart from basic facts, significant
    information should not appear in the lead if
     it is not covered in the remainder of the
    article.

Generate the lead for the article titled "{
    page_title}" using the article's body above
    with no more than {n_toks_trgt} words.
    Article:

"""{src}"""
```

B.3 TST

B.3.1 Prompts

Minimal

```
Please make this sentence/paragraph more neutral.
     **Make as few changes as possible and use
    no more than {trgt_n_words} words for the
    neutralised sentence/paragraph.** Sentence/
    Paragraph:

"""{src}"""
```

Instruction/Few-Shot

```
Please edit this biased Wikipedia sentence/
    paragraph to make it more neutral, aligning
    with Wikipedia's neutral point of view
    policy:

Achieving what the Wikipedia community
    understands as neutrality means carefully
    and critically analyzing a variety of
    reliable sources and then attempting to
    convey to the reader the information
    contained in them fairly, proportionately,
    and as far as possible without editorial
    bias. Wikipedia aims to describe disputes,
    but not engage in them. The aim is to inform
    , not influence. Editors, while naturally
    having their own points of view, should
    strive in good faith to provide complete
    information and not to promote one
    particular point of view over another. The
    neutral point of view does not mean the
    exclusion of certain points of view; rather,
     it means including all verifiable points of
     view which have sufficient due weight.
    Observe the following principles to help
    achieve the level of neutrality that is
    appropriate for an encyclopedia:

- Avoid stating opinions as facts.
- Avoid stating seriously contested assertions
    as facts.
- Avoid stating facts as opinions.
- Prefer nonjudgmental language.
- Do not editorialize.
- Indicate the relative prominence of opposing
    views.

**Make as few changes as possible and use no
    more than {trgt_n_words} words for the
    neutralised sentence/paragraph.** Output
    only the neutralized sentence/paragraph.
    Sentence/Paragraph:

"""{src}"""
```

B.3.2 Style Classifiers

We fine-tune four style classifiers: one for each language at the sentence level, and an additional classifier for English at the paragraph level. The hyperparameter settings are provided in Table 8.

Language/Level	Models	Learning Rate	Batch Sizes	Epochs	Weight Decay
EN/Sent.	roberta-base	1e-6	32	15	0.01
PT/Sent.	xlm-roberta-base, mBERT	5e-5, 1e-5, 5e-6	16, 32	2, 5, 8	0, 0.01
VI/Sent.	xlm-roberta-base, mBERT	5e-5, 1e-5, 5e-6, 1e-6	16, 32	2, 4, 6	0, 0.01
EN/Para.	roberta-base	5e-5, 1e-6, 5e-6	16, 32	3, 6, 9	0, 0.01

Table 8: Style Classifier Hyperparameter Settings.

For English, we adopt the hyperparameters from the best-performing neutrality classifier available on Hugging Face.[34] As the English data contain nearly a quarter of a million sentence pairs, we fine-tune on a smaller subset of the most recent 150k pairs, specifically filtered to include the keyword *NPOV* in the revision content, in order to further enhance precision. For Portuguese, we apply commonly used hyperparameter values, while for Vietnamese and English paragraphs, we extend the search space, as initial experiments yielded low detection performance.

Level	Language	Pairs	Test Accuracy
Sentences	English	300,000	73%
	Portuguese	5738	63%
	Vietnamese	2370	58%
Paragraphs	English	9342	58%

Table 9: Style Transfer Classifier Performance. Pairs denote biased and neutralised samples.

Table 9 reports the style classifier hyperparameter fine-tuning results. While fine-tuned models for English and Portuguese sentences yield satisfactory results, style accuracy for English paragraphs and Vietnamese sentences is low. In the following, we provide a qualitative analysis of both subsets and explain how we address these low performances.

Low Style Classifier Performance Analysis Table 10 presents two representative examples of NPOV revisions from each subset. The first example in each case illustrates a clear NPOV violation. For instance, the phrase "considered the

[34] https://huggingface.co/cffl/bert-base-styleclassification-subjective-neutral

best footballer" in Vietnamese and "not as strong" in English are both subjective. However, as illustrated with the second examples, NPOV filtering also captures revisions related to political or historical content, which often rely on (subjectively) factual corrections rather than systematic semantic cues.

Subset	Biased Examples
Vietnamese	***Đc coi là cu th xut sc nht th gii và là cu th vĩ đi nht mi thi đi (Greatest of All Time - GOAT)**, Ronaldo là ch nhân ca 5 Qu bóng vàng châu Âu vào các năm 2008, 2013, 2014, 2016, 2017 và cũng là ch nhân 4 Chic giày vàng châu Âu, c hai đu là k lc ca mt cu th châu Âu cùng nhiu danh hiu cao quý khác.* (EN: Considered the best football player in the world and the greatest of all time (GOAT), Ronaldo has won 5 Ballon d'Or awards in the years 2008, 2013, 2014, 2016, and 2017, as well as 4 European Golden Shoes—both records for a European player—along with many other prestigious titles.)
	*Ông tng phc v Lý Hoài Tiên, tng di quyn **nghch tc** S T Minh ca Ngy Yên.* (EN: He once served Lý Hoài Tiên, a general under the command of the rebel S T Minh of Ngy Yên.)
English Paragraphs	*He is **not as strong**, although still an exceptional warrior. Agamemnon clearly has a stubborn streak that **one can argue makes him even more arrogant** than Achilles. Although he takes few risks in battle, Agamemnon **still accomplishes great progress** for the Greeks.*
	*The population of Bangladesh ranks seventh in the world, but its area of approximately is ranked **ninety-fourth**, making it one of the most densely populated countries in the world, or the most densely populated country if small island nations and city-states are not included. It is the third-largest Muslim-majority nation, **but** has a smaller Muslim population than the Muslim minority in India. Geographically **dominated** by the fertile Ganges-Brahmaputra Delta, the country has annual monsoon floods, and cyclones are frequent.*

Table 10: NPOV Revision Examples. Parentheses contain English translations. Highlighted words indicate words that were edited.

As we observed this pattern consistently across both subsets, we conducted additional data processing and hyperparameter tuning for the classifiers. We explored several strategies, including: (1) extending the list of NPOV-related keywords, (2) allowing multiple edit chunks per revision, (3) permitting multi-sentence edits within a single chunk, and (4) expanding the range of hyperparameter settings and model types. However, none of these approaches significantly improved style classifier performance.

Therefore, we selected the configuration that yielded the highest precision, adopting a conservative approach to extract NPOV-relevant revision pairs. Despite the relatively low classifier accuracy, we are confident that our dataset includes a high proportion of true positives.

C Detector Details and Implementations

We follow the taxonomy for detecting MGT proposed by (Yang et al., 2023), which categorises detectors into three types: 1) zero-shot, 2) training-based, and 3) watermarking, although we exclude the latter from our experiments. The taxonomy further divides zero-shot methods into white-box and black-box, depending on whether the detector has access to the generator's logits or other model internals. For all detectors, when the original baseline LLM does not support one of our languages, we replace it with a multilingual model of comparable size. For zero-shot detectors, we use Youden's J statistic to determine the optimal threshold.

Zero-shot White-box

LLR (Su et al., 2023) The Log-Likelihood Log-Rank Ratio (LLR) intuitively leverages the ratio of absolute confidence through log-likelihood to relative confidence through log rank for a given sequence. We implement this detector with Bloom-3B.[35]

Binoculars (Hans et al., 2024) Binoculars introduces a metric based on the ratio of perplexity to cross-perplexity, where the latter measures how surprising the next-token predictions of one model are to another. We implement this detector using Qwen2.5-7B[36] for the observer model and Qwen2.5-7B-Instruct[37] for the performer model.

FastDetectGPT White-Box (Bao et al., 2023) DetectGPT (Mitchell et al., 2023) exploits the observation that MGT tends to be located in regions of negative curvature in the log-probability function, from which a curvature-based detection criterion is defined. FastDetectGPT (WB) is an optimised version of DetectGPT that builds on the *conditional*

[35] https://huggingface.co/bigscience/bloom-3b
[36] https://huggingface.co/Qwen/Qwen2.5-7B
[37] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Hyperparameter	Values
Batch Size	16, 32
Learning Rate	1e-5, 5e-6, 1e-6
Epochs	3, 5

Table 11: Hyperparameter settings for supervised-detectors.

probability curvature. We implement the white-box version with Bloom-3B.[35]

Zero-shot Black-box

Revise (Zhu et al., 2023a) Revise builds on the hypothesis that ChatGPT[38] performs fewer revisions when generating MGT, and thus bases its detection criterion on the similarity between the original and revised articles. We implement this detector as in the original paper, using GPT-3.5-turbo.[39]

GECScore (Wu et al., 2025) Grammar Error Correction Score assumes that HWT contain more grammatical errors and calculates a Grammatical Error Correction score. We implement this detector as in the original paper, using GPT-3.5-turbo.[39]

FastDetectGPT Black-Box (Hans et al., 2024) In the black-box version, the scoring model differs from the reference model. We use BLOOM-3B as the reference model and BLOOM-1.7B as the scoring model.

Supervised

XLM-RoBERTa (Conneau et al., 2020): XLM-RoBERTa[40] is the multilingual version of RoBERTa (Liu et al., 2019) for 100 languages. RoBERTa improves upon BERT (Devlin et al., 2019) through longer and more extensive training, as well as dynamic masking.

mDeBERTaV3 mDeBERTaV3[41] is the multilingual version of DeBERTa (He et al., 2023), which enhances BERT and RoBERTa using disentangled attention and an improved masked decoder.

[38] https://openai.com

[39] https://platform.openai.com/docs/models/gpt-3.5-turbo

[40] https://huggingface.co/FacebookAI/xlm-roberta-base

[41] https://huggingface.co/microsoft/mdeberta-v3-base

Both models are fine-tuned per task and language on an 80/10/10 split with the hyperparameter choices displayed in Table 11.

D Additional Results

D.1 Mistral Error Analysis

We observe anomalous evaluation metrics for Vietnamese texts written by Mistral. While both zero-shot detectors achieve random chance accuracy and often zero F1-scores, training-based detectors achieve near-perfect metrics. Upon inspecting the data, we find that Mistral, unlike the other models, fails to follow the instructions in our prompts. Common errors include outputting text mid-sentence or returning English text, despite the final sentences of our prompts emphasising that the response should be in Vietnamese. These flaws explain the strong performance of training-based detectors, as they detect such syntactic imperfections, whereas zero-shot detectors appear unable to identify clear patterns based on model internals or token-level features.

E Paper Checklist

Benefits

Q1 *How does this work support the Wikimedia community?*

A1 We believe our work supports the Wikimedia community in at least two ways. First, we introduce two new text corpora that extend beyond MGT detection and can be leveraged for various AI applications. The mWNC dataset addresses (1) community requests to expand existing resources with additional languages, and (2) high-priority needs identified by workshop organizers for NPOV datasets to train and evaluate models for biased language detection. These data open up several research directions, such as training models to detect bias in longer text sequences and testing their generalisability across varying text lengths. As our style classifier results suggest, NPOV detection in low-resource languages remains challenging, making mWNC a valuable resource for advancing this area of research.

Likewise, with WikiPS, we aim to provide two large-scale subsets of general interest to the research community. The paragraph-level subset, for instance, can be used to build question answering datasets for non-high-resource languages, analogous to SQuAD (Rajpurkar et al., 2016). Our

summarisation subset naturally lends itself to improving lead section summarisation models. We highlight the inclusion of infoboxes as a key input feature for lead generation, in line with recent findings that LLM-generated summaries are often on par with—or even preferred over—human-written ones (Goyal et al., 2022; Pu et al., 2023; Zhang et al., 2024).

Second, our benchmark, WETBench, is designed to inform the Wikipedia community about the feasibility and effectiveness of current state-of-the-art detectors in identifying MGT instances on the platform. As outlined in the introduction, there is growing concern about the influx of low-quality, unreliable machine-generated content. Due to limitations in prior evaluations (see Section 1), we hope our work contributes to a better understanding of the capabilities and limitations of current detectors, supporting future research and real-world efforts to identify and manage MGT on Wikipedia.

Q2 What license are you using for your data, code, models? Are they available for community re-use?

A2 We release our datasets, WikiPS and mWNC, which are derived from Wikipedia, under the CC BY-SA 4.0 license. Users of the MGT included in our benchmark must ensure compliance with the respective licenses of each language model (see Ethics Statement). We open-source all code used in our work.

Q3 Did you provide clear descriptions and rationale for any filtering that you applied to your data? For example, did you filter to just one language (e.g., English Wikipedia) or many? Did you filter to any specific geographies or topics?

A3 We provide comprehensive explanations of our dataset construction in Section 3 and Appendix A. Section 3 outlines the high-level construction process and key design choices, while Appendix A offers a detailed walkthrough for readers interested in replicating or closely examining our methodology. For fine-grained construction details, we refer readers to our publicly available codebase.

Risks

Q1 If there are risks from your work, do any of them apply specifically to Wikimedia editors or the projects?

A1 Our research objective is to provide a more accurate assessment of SOTA MGT detectors' performance on task-specific MGT. We acknowledge that our findings could be misinterpreted or misused to claim that SOTA detectors are ineffective at identifying machine-assisted edits. However, the intent of our work is not to undermine the potential of detection methods but to highlight their current limitations in realistic editorial settings.

Q2 Did you name any Wikimedia editors (including username) or provide information exposing an editor's identity?

A2 No. Our data includes only textual information, without any references to individual editors.

Q3 Could your research be used to infer sensitive data about individual editors? If so, please explain further.

A3 No. While our dataset includes revision IDs, it does not contain any additional information that is not already publicly available on Wikipedia.