XDAC: XAI-Driven Detection and Attribution of LLM-Generated News Comments in Korean

Wooyoung Go[1,2] **Hyoungshick Kim**[3] **Alice Oh**[2] **Yongdae Kim**[2]

[1]National Security Research Institute [2]KAIST [3]Sungkyunkwan University

gwy876@nsr.re.kr, hyoung@skku.edu, alice.oh@kaist.edu, yongdaek@kaist.ac.kr

Abstract

Large language models (LLMs) generate human-like text, raising concerns about their misuse in creating deceptive content. Detecting LLM-generated comments (LGC) in online news is essential for preserving online discourse integrity and preventing opinion manipulation. However, effective detection faces two key challenges: the brevity and informality of news comments limit traditional detection methods, while the lack of publicly available LGC datasets hinders model development, particularly for non-English languages. To address these challenges, we propose a twofold approach. First, we develop an LGC generation framework to construct a high-quality dataset with diverse and complex examples. Second, we introduce XDAC (**X**AI-Driven **D**etection and **A**ttribution of LLM-Generated **C**omments), a framework utilizing explainable AI, designed for the detection and attribution of short-form LGC in Korean news articles. XDAC leverages XAI to uncover distinguishing linguistic patterns at both token and character levels. We present the first large-scale benchmark dataset, comprising 1.3M human-written comments from Korean news platforms and 1M LLM-generated comments from 14 distinct models. XDAC outperforms existing methods, achieving a 98.5% F1 score in LGC detection with a relative improvement of 68.1%, and an 84.3% F1 score in attribution. To validate real-world applicability, we analyze 5.24M news comments from Naver, South Korea's leading online news platform, identifying 27,029 potential LLM-generated comments.

1 Introduction

State-of-the-art large language models (LLMs) generate text that closely mimics human writing, raising concerns about AI-generated misinformation. Among various forms of AI-generated content, *news comments* are particularly problematic due to their influence on public opinion and ease of manipulation (Jiang and Wilson, 2018; Kim and Masullo Chen, 2021; Zerback and Töpfl, 2022). Unlike traditional bot-generated comments, LLM-generated comments (LGC) exhibit human-like fluency, making them harder to detect (Luceri et al., 2024; Feng et al., 2024; Wan et al., 2024). This threat to information integrity highlights the urgent need for reliable detection methods.

Existing research on LLM-generated text detection has primarily focused on long-form content such as articles or essays (Solaiman et al., 2019a; Kumari et al., 2023; Zhong et al., 2020; Kumarage et al., 2023a; Gehrmann et al., 2019). However, these methods struggle with *short-form, informal text like news comments*, which often lack sufficient lexical and syntactic complexity for traditional detection techniques to be effective (Kumarage et al., 2023b; Mitrović et al., 2023; Bao et al., 2023; Solaiman et al., 2019b; Gameiro et al., 2024). Common LLM detection tools, such as GPTZero,[1] impose length constraints (e.g., a minimum of 250 characters), which makes them unsuitable for detecting LGC in real-world settings. Our analysis of Korean news comments reveals an average length of 51 characters (11 words), highlighting the significant gap between the requirements of existing tools and the characteristics of real-world comments. Furthermore, since these approaches primarily depend on word probability distributions or stylometric features, their effectiveness diminishes considerably when applied to short, casual expressions.

A key factor compounding the difficulty of short-form LGC detection is the absence of realistic training data. While LLMs can generate synthetic comments, naïve generation often results in repetitive or easily identifiable outputs, failing to capture the nuanced variability of human-written comments (HWC). This lack of realistic data makes it challenging to train models that effectively distinguish

[1]https://gptzero.me

LGC from HWC, especially in short-form.

To address these challenges, we introduce **XDAC** (**X**AI-Driven **D**etection and **A**ttribution of LLM-Generated **C**omments), a framework specifically designed for both generating realistic LGC and subsequently detecting and attributing them. Our work focuses on two key tasks:

- **LGC Detection:** Determining whether a comment is HWC or LGC.
- **LLM Attribution:** Identifying the specific LLM responsible for generating a given LGC.

We tackle these challenges through two key strategies. First, to create a realistic LGC dataset, we develop a sophisticated LGC generation framework that mitigates the limitations of naïve LLM prompting. XDAC employs diverse LLMs, enhances comment naturalness (incorporating informal language, emojis, and emotional expressions), provides fine-grained sentiment control, and uses reference-augmented generation. Second, to address the inherent challenges of short-form text analysis, we leverage XAI. Recognizing that traditional methods struggle with short, informal content, we utilize XAI to uncover subtle stylistic and linguistic patterns that distinguish LGC from human-written content. Our analysis reveals distinct LGC characteristics, such as a preference for formal structures and standardized expressions (e.g., "것 같다" ("it seems")) while lacking informal elements (e.g., repeated characters, emotional symbols like "ㅋㅋㅋㅋㅋ" ("LOL")). These XAI-driven insights directly inform our robust, short-form LGC-optimized detection model.

XDAC achieves 98.5% F1 in LGC detection and 84.3% in LLM attribution, outperforming existing methods. To validate its real-world applicability, we analyzed 5.24M news comments posted on Naver, a leading Korean news platform, between January 2023 and August 2024, identifying 27,029 potential LGC. Upon camera-ready, we will publicly release our source code and dataset to facilitate future research in LGC detection (https://github.com/airobotlab/XDAC).

2 Background

Misuse of LLMs: The misuse of LLMs is widespread across various domains, raising significant societal concerns. It includes the generation of fake news (Ahmed et al., 2021; Hacker et al., 2023; De Angelis et al., 2023; Zellers et al., 2019), malicious product reviews (Adelani et al., 2020; Abdelnabi and Fritz, 2021), and misleading social media posts (Shu et al., 2018; Fagni et al., 2021a), all of which contribute to harm and confusion. Additionally, LLMs are exploited to manipulate public opinion (Spitale et al., 2023; Goldstein et al., 2024; Lucas et al., 2023; Chen and Shu, 2023; Goldstein et al., 2023; Buchanan et al., 2021) through the dissemination of mis/disinformation and propaganda. These actions collectively erode public trust and threaten the foundations of democratic systems.

LLM-Generated Text Detection: To address the above-mentioned issues, researchers have developed various LLM detection methods. LM-based approaches (Solaiman et al., 2019a; Zellers et al., 2020; Uchendu et al., 2021a; Fagni et al., 2021b; Liu et al., 2024; Pu et al., 2023; Uchendu et al., 2021b) leverage pretrained language models such as BERT and RoBERTa. The feature-augmented approach encompasses techniques such as energy-based (Kumari et al., 2023), structural (Zhong et al., 2020; Gambini et al., 2023; Liu et al., 2023), and stylometry (Kumarage et al., 2023a; Mindner et al., 2023; Mikros et al., 2023; Kumarage et al., 2023c) methods. There are also zero-shot methods for identifying LLM-generated texts without the need for additional training (Gehrmann et al., 2019; Su et al., 2023; Mitchell et al., 2023; Wang et al., 2023b; Bao et al., 2023; Wang et al., 2023a; Guo and Yu, 2023). There are MGT (Machine-Generated Text) services available online, including GPTZero, zerogpt,[2] ai-content-detector,[3] ai-detector[4].

Unlike prior work, which often focuses on longer English texts, our research uniquely tackles the challenges of detecting LLM-generated short-form Korean comments prevalent in online news. In doing so, we first generate a large, diverse dataset of synthetic comments that mimic the naturalness of the HWC by incorporating sentiment and writing styles. Then we build a more effective detection system, XDAC. Our system includes a comprehensive analysis of linguistic patterns and features unique to LGC, leveraging XAI techniques, and also explores LLM attribution to identify specific generative models. We conduct extensive real-world testing to validate the effectiveness of our model.

[2] https://www.zerogpt.com
[3] https://writer.com/ai-content-detector
[4] https://www.scribbr.com/ai-detector

3 XDAC

3.1 Overview of XDAC

Figure 1 shows the architecture of XDAC. The framework consists of three main components: “LGC generation,” “linguistic patterns extraction,” and “detection and attribution.”

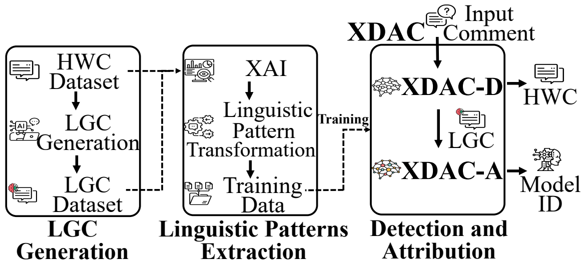


Figure 1: Overview of XDAC.

During the LGC generation phase, we created a dataset of 1.3M publicly available Korean HWC collected from various news platforms. We then generated 1M Korean LGC using 14 state-of-the-art LLMs. To ensure authenticity and diversity, we implemented strategies to enhance comment realism, incorporate sentiment variations, and utilize reference-driven generation techniques.

For linguistic pattern extraction, we employ XAI techniques, including Guided GradCAM (Selvaraju et al., 2017) and CAPTUM (Kokhlikyan et al., 2020), to identify key distinctions between LGC and HWC. LGC exhibits formal structures and standardized expressions, whereas HWC shows greater variation in informal and emotional expressions. We also observe notable differences in special character usage, formatting, and repetition. Leveraging these insights, we develop a linguistic feature-based tokenizer to encode LGC-specific characteristics, enhancing detection accuracy by capturing distinctive tone, formatting, and linguistic features identified through XAI analysis.

XDAC operates in two modes: XDAC-D for LGC detection (HWC vs. LGC) and XDAC-A for attribution (identifying the generating LLM).

3.2 LGC Generation

A high-quality dataset comprising both LGC and HWC is essential for developing a robust LGC detection model. This section details the dataset construction process, including HWC collection and LGC generation. The HWC dataset, pivotal for our LGC generation framework and XDAC model training, contains 1.3M comments from 135K news articles. These articles were collected from major Korean news platforms via official APIs in 2022, prior to November, before the public release of ChatGPT, to minimize LLM content. We collected the most commented articles (with 15 or more comments) and carefully constructed the dataset by extracting comments while eliminating duplicates, as well as deleted and hidden entries. An additional 5.2M comments from 2023–2024 were exclusively reserved for real-world deployment and analysis (Section 5.3) to investigate potential LLM generation. All HWC ranged from 15–280 characters or 4–50 words. Using the LGC generation framework, we created 1M Korean LGC that reflect the stylistic and contextual characteristics of real news comments. Detailed statistics for both HWC and LGC are provided in Appendix A.

3.2.1 LGC Generation Framework

We developed the LGC generation framework to generate human-like comments that are difficult for human readers to distinguish from HWC. Directly prompting an LLM to produce comments, especially those with specific sentiments like negativity, often results in limited diversity, repetitive outputs, and may trigger LLM safety guardrails, leading to generic or refused responses. To address these limitations, our framework simulates a broad range of scenarios and strategic variations from an adversarial perspective. This involves meticulously selecting diverse LLM models to broaden linguistic styles, enhancing comment naturalness by mimicking human writing traits, enabling fine-grained sentiment control beyond simple positive/negative, and leveraging news content and existing human comments to ensure high relevance and contextual nuance. These combined strategies collectively ensure the generation of diverse, contextually nuanced, and less detectable LGC. The framework is structured around four key components.

❶ **LLM Model Selection:** Selecting appropriate LLMs is crucial as they directly influence the linguistic style, complexity, and emotional tone of the generated comments. Our framework employs 14 state-of-the-art LLMs, balancing API-based and open-source models for optimal generation quality and computational feasibility. We included HCX and Bllossom, models exclusively or primarily trained for Korean, as well as high-performing generalist models like OpenAI’s GPT series, Google’s Gemini and Gemma-2, Anthropic’s Claude 3.5 Sonnet, and Alibaba Cloud’s Qwen2.5. These models were selected for their effectiveness in generating diverse Korean LGC, even if their primary language is English or Chinese (Appendix B.1).

❷ **Enhancing Comment Naturalness:** To ensure LGC closely resemble genuine user input, we apply six strategies that mimic human writing, all incorporated into the generation prompt. These include using informal, conversational language, embedding emojis, introducing minor textual variations, expressing emotions naturally, using special characters for emphasis, and keeping comments concise. These techniques collectively help produce more realistic and engaging content (Appendix B.2).

❸ **Sentiment Subtype Selection:** Our framework enables fine-grained sentiment control by selecting from 32 positive and 37 negative sentiment subtypes, including a None subtype. This approach surpasses simple binary classification (e.g., positive, negative, neutral) to allow for nuanced tonal adjustments, ensuring generated comments reflect a contextually appropriate stance on news discussions. These detailed subtypes were meticulously derived from an analysis of sentiment expressions in HWC, enhancing the diversity and realism of generated LGC, and accurately capturing varied perspectives (Appendix B.3).

❹ **Reference-Augmented Generation Strategies:** The framework employs four generation methods based on reference usage: *Direct Generation*, *News-based generation*, *News and comment-based generation*, and *Opinion-based generation*. These approaches enable the generation of diverse comments with varying levels of contextual relevance and alignment with user perspectives (Appendix B.4).

Our LGC generation prompt, which incorporates these strategies, is provided in Appendix B.5. During generation, 11.8% of prompts were blocked or failed due to LLM guardrails.

3.2.2 Model-Based Evaluation of LGC

Evaluating LGC requires a tailored approach due to the lack of established benchmarks and the impracticality of constructing new prompt-comment pairs. We adopt the LLM-as-a-Judge (Zheng et al., 2024; Kim et al., 2024a,b; Fu et al., 2023) framework, leveraging *GPT-4o-2024-08-06* as the primary evaluator given its strong performance in Korean text assessment. Details regarding the dataset, evaluation criteria, and prompt templates are provided in Appendix C.

Quality Evaluation. Table 1 presents the results of quality evaluation, focusing on specificity and fairness. In general, larger LLMs yielded higher specificity scores. For fairness, models such as *HCX-003* and *Llama3-Blossom-70B* received lower ratings, likely due to their strict adherence to the sentiment-controlled *LGC Generation Framework*.

Evaluation Type	Quality Assessment		Prompt Reflection		
model	**Specificity**	**Fairness**	**Content Relevance**	**Authenticity Reflection**	**Sentiment Reflection**
gpt-3.5-turbo-0125	63.0	90.1	96.5	77.3	87.9
gpt-4-0125-preview	74.0	91.9	98.6	81.5	94.9
gpt-4o-2024-05-13	62.8	90.1	96.2	82.0	96.5
gpt-4o-2024-08-06	68.0	93.8	95.8	79.7	95.7
claude-3-5-sonnet	66.0	91.0	95.5	84.7	97.8
gemini-pro	66.4	87.8	99.0	80.2	80.4
HCX-DASH-001	55.0	94.3	97.9	68.5	62.5
HCX-003	56.5	84.9	99.0	79.7	77.9
gemma-2-9b-it	42.5	91.4	94.8	84.4	84.9
gemma-2-27b-it	52.1	88.8	97.6	90.1	86.7
Qwen2.5-7B-Instruct	40.9	93.5	96.2	74.0	72.7
Qwen2.5-32B-Instruct	40.9	89.8	94.1	84.1	89.6
llama-3-Bllossom-8B	46.4	87.8	96.5	74.5	82.3
llama-3-Bllossom-70B	51.0	81.0	97.2	83.6	85.2
Total	**55.9**	**89.7**	**96.8**	**80.2**	**85.1**

Table 1: LGC evaluation by LLM-as-a-Judge.

Prompt Reflection Evaluation. The same table reports how faithfully LLMs followed prompt instructions. We assessed three criteria: (1) content relevance—alignment with the news article, (2) authenticity reflection—human-likeness in expression, and (3) sentiment reflection—alignment with the specified sentiment subtype. While content relevance was consistently high across models, larger models performed better in terms of authenticity and sentiment reflection. Appendix C.1 provides further analysis. Notably, positive sentiment prompts were correctly reflected in 84.4% of cases, while negative ones showed lower alignment at 67.2%. Some models demonstrated sentiment bias, with *Qwen2.5-7B* favoring positive sentiment and *Claude-3-5-Sonnet* frequently producing positive-toned comments regardless of the prompt.

3.2.3 Human Evaluation of LGC

To assess the quality and human-likeness of generated comments, we conducted a human evaluation study comparing LGC with HWC. Following established protocols for machine-generated text evaluation (He et al., 2023), four adult evaluators independently rated 210 comments (140 LGC, 70 HWC), each paired with its corresponding news article. Comments were evaluated across six dimensions using a 3-point Likert scale: LLM authorship perception, relevance, specificity, fairness, fluency and naturalness, and sentiment. Evaluators were blinded to the origin of each comment. Detailed annotation procedures and evaluation results are provided in Appendix D. Most LGC (67.1%) were perceived as human-written, compared to 72.9% for HWC, demonstrating high indistinguishability. LGC achieved superior performance in relevance (94.8% for LGC vs. 87.1% for HWC) and fluency (71.3% for LGC vs. 44.6% for HWC), while specificity ratings were slightly higher for LGC (49.5% for LGC vs. 41.8% for HWC). HWC were more frequently perceived as biased (50.0% for HWC vs. 33.2% for LGC). In terms of sentiment, LGC exhibited a more balanced distribution, with higher positive sentiment (48.2%) and lower negativity (37.9%) compared to HWC (10.0% positive, 77.9% negative). These results indicate that LGC often matches or exceeds HWC in terms of contextual appropriateness and fluency while maintaining greater neutrality. Furthermore, human and model-based evaluations showed consistent alignment in relevance and specificity ratings, supporting the validity of our LLM-as-a-Judge approach for Korean comment evaluation.

3.3 Linguistic Patterns Extraction

This section analyzes LLM-generated Korean news comments. XAI techniques were essential for understanding the stylistic differences between HWC and LGC and identifying their specific patterns.

XAI-Driven Linguistic Analysis: We applied XAI techniques to analyze linguistic differences between LGC and HWC by fine-tuning a 1D CNN model with Guided GradCAM (Zhou et al., 2016; Selvaraju et al., 2017; Go and Lee, 2018) and a KCBERT model using Captum (Kokhlikyan et al., 2020). These models achieved F1 scores of 93.1 and 95.3, respectively, by identifying the most influential comment segments for classification. For explainability, we used Captum's integrated gradients method, specifically layer-integrated gradients, to compute feature attributions. This approach involves tokenizing the input text, calculating attributions for each token by integrating gradients from a baseline and mapping these attributions back to the character level. After filtering out special tokens like `[CLS]`, `[SEP]`, and `[PAD]`, we aggregated token-level attributions and detected key linguistic patterns based on sequences with high attribution scores. This process enabled us to identify which input components most significantly influenced the model's predictions, providing a clear understanding of how the model distinguishes between LGC and HWC. Training details can be found in Appendix E.1

We identified characteristic linguistic patterns by analyzing 80,000 comments from the training set only. This analysis revealed 200 frequently occurring patterns for each group (LGC and HWC). In human comments, these patterns appeared 65,435 times, with 12,616 marked as key by XAI, while in LLM comments, the patterns appeared 182,310 times, with 90,909 highlighted as key. These stylistic patterns were context-dependent and not always critical in every instance. LLM-specific phrases are often repeated within the same comment. Appendix E.2 presents the frequency distribution of key stylistic patterns for LGC and HWC identified by XAI in the KcBERT detection model.

For LGC, the XAI results revealed a tendency to rely on formal, structured phrases with standardized expressions. Common phrases such as "것 같다" ("it seems"), "에 대해" ("about"), along with frequent connectors, were consistently identified as high-impact regions in the model's decision-making process. In contrast, human-generated comments demonstrated a greater variety of informal expressions, including emotional symbols such as "ㅋㅋㅋㅋㅋ" ("LOL" or laughing), "...." (used to indicate hesitation or trailing off), and context-specific terms, which were identified as high-influence linguistic features. Human comments also exhibited more frequent use of personal pronouns, emotive phrases, and culturally specific language compared to LGC.

This XAI analysis reveals fundamental differences in language patterns between LGC and HWC. While LLMs consistently produce standardized, neutral text, human comments exhibit greater linguistic variety and emotional expressiveness. These findings provide valuable insights for enhancing LLM architectures to more accurately capture the natural variations and nuances characteristic of human writing.

Profiling LLM-Specific Styles: Our analysis revealed distinct stylistic differences across different LLMs. For example, GPT-4 tends to generate comments with a formal tone and precise grammar, while LLaMA-produced comments exhibited a more conversational style. These profiles were used to further refine our detection models.

Special Character Usage: Figure 2 shows special character usage by source. Despite emojis being intentionally included in LGC to align with human-like patterns, LGC exhibit a distinct, standardized usage relying on globally recognized symbols, often absent in HWC, thus limiting diversity and cultural nuance. In contrast, HWC displays more varied and context-specific usage, reflecting greater stylistic and cultural depth.

Special characters used only by **humans**	Special characters used exclusively by **LLMs**
‥, ・, ㅈ, ㄱ, 🐕, ㅅ, ㄹ, ㄴ, ㅂ, ■, ♡, 💙, ㅊ, ″, ＼, “, ㅁ, `, ㄲ, ×, 、, ㅆ, A, ㅌ, ●, ★, ㅍ, 🖕, ㅃ, •, 🍉, →, 👉, ₩t, ₩x00, ₩u200b, —, —	🤔, 😊, 💪, 😠, 😔, 🌟, 🎉, ✨, 😳, 🙄, 😄, 💖, 😰, 😒, 🙌, 🧐, ♂, 🙁, 😥, 🤩, 😱, 💯, 🌈, 😞, ♀, 🤷, 🚀, 🤝, 💡, 🤦, 🥺, 🤯, 👀, 😅, 😩, 😓, 🥰, 💔, 💕, 📚

Figure 2: Special character usage patterns.

Formatting Character Usage: Formatting character usage differs significantly between LGC and HWC. As shown in Table 2, only 0.001% of LGC contain newlines, and double spaces are rare, whereas 26.1% of HWC use formatting characters, with significant usage of double spaces (19.1%) and newlines (10.2%). This minimal formatting in LGC can be attributed to the preprocessing policies of LLM training data.

Pattern Type	**HWC (%)**	**LGC (%)**
Double space	19.1	1.1
Newline	10.2	0.001
Double Newline	0.8	0.0
TAB	0.0001	0.0
All	26.1	1.1
Repeated Characters	**HWC (%)**	**LGC (%)**
$\geq$ **2**	51.69	11.61
$\geq$ **3**	22.90	4.09
$\geq$ **4**	8.18	0.09
$\geq$ **5**	3.89	0.04

Table 2: Formatting and repetition patterns comparison.

Repeated Character Usage: LGC rarely use repeated characters, likely due to preprocessing and repetition penalties, resulting in standardized text. In contrast, HWC frequently employ them for emotion and emphasis. As shown in Table 2, 51.69% of HWC contain repeated characters, whereas only 11.61% of LGC do, with the gap widening as repetitions increase.

3.4 Linguistic Feature-Based Tokenizer

Our analysis of LGC and HWC using XAI techniques revealed several key linguistic patterns. These include unique tones for each LLM, distinct special character usage, and differences in formatting and repetition patterns. HWC tends to contain more formatting characters (such as multiple spaces or line breaks) and repetitive characters, while LGC often lacks these features. Leveraging these insights, we designed a specialized tokenizer that incorporates these nuanced linguistic features to optimize our detection model, enabling more precise identification of LGC.

Conventional subword tokenization methods, such as BERT-based WordPiece tokenizers, fail to capture essential linguistic features that distinguish LGC from HWC. These tokenizers face two main limitations: they struggle to handle repetitive characters effectively, and they process formatting elements (spaces, newlines, tabs, and their multiple repetitions) as mere separators, losing the semantic significance that is more prevalent in human-written text. Moreover, these methods fail to account for the unique tones of different LLMs and their specific patterns of special character usage. This inadequacy often results in incomplete tokenization, limiting the effectiveness of traditional detection approaches when applied to LGC.

To overcome these limitations, we propose XDAC, a tokenizer designed to handle these nuanced linguistic features. XDAC incorporates tone tokens from XAI analysis and effectively processes repetitive patterns, spaces, and formatting characters, significantly improving detection accuracy compared to baseline models like LM-D.

Incorporation of Formatting and Special Character Tokens: We enhance the tokenizer by adding formatting tokens (e.g., “<SPACE>” for a space character, “<ENTER>” for a newline, and “<TAB>” for a tab character), which help the model capture formatting patterns, and 560 commonly used special characters from both LLM and human texts. This expansion minimizes the use of unknown tokens, improving the model’s ability to distinguish between LGC and HWC.

Repetitive Pattern Transformation: We introduce a transformation module that explicitly encodes repetition, as described in Appendix E.3. Using new tokens “<REP>” and “</REP>” for improved precision indicating the start and end of a repetition, repetitive sequences are effectively captured. For example, “ㅋㅋㅋㅋㅋ” is transformed into “<REP> ㅋ 5 </REP>” to capture both the character and repetition count. This approach also applies to spaces (“   ”) and line breaks (“\n\n”), encoding them as “<REP> <SPACE> 5 </REP>” and “<REP> <ENTER> 2 </REP>”, preserving repetitive patterns without loss of meaning.

Inclusion of Tone Tokens from XAI Analysis: We incorporate 300 special tokens derived from XAI analysis to capture frequently used phrases characteristic of both LLMs and human writers. Common LLM expressions (e.g., “것 같아”

(“seems like”), “는 것은” (“the fact that”)) and human expressions (e.g., “고...” (“as well as...”), “국회의원” (“Member of the National Assembly”)) are added to the tokenizer’s vocabulary (Appendix E.4). These tokens enhance the model’s ability to detect stylistic differences between LGC and HWC with greater accuracy. Based on the three approaches mentioned above, we designed the input transformation for XDAC to capture various characteristics specific to LGC. Figure 3 presents an example illustrating the differences in input sentence transformation between the existing methods and the proposed method.

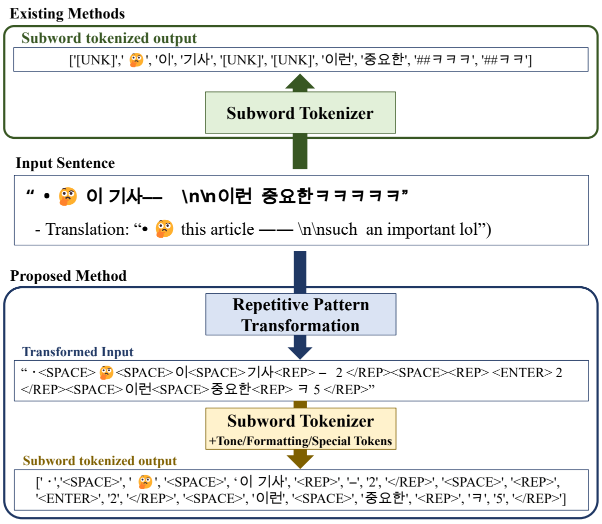


Figure 3: Comparison between existing methods and the proposed method for input sentence transformation.

4 Experimental Setup

This section provides an overview of the experimental setup, detailing the datasets as well as the models and implementation procedures, which are further described in Appendix F. To evaluate performance by text length, and given our dataset primarily consists of inherently short comments (LGC average 11 words), we categorize these test data into L-Text (short for Long Text, Words > 12), M-Text (Medium Text, Words 6–12), and S-Text (Short Text, Words < 6), enabling performance assessment on extremely short comments. Throughout the results, Total is reported as F1, HWC as TNR, and LGC as TPR.

4.1 LGC Detection Models

We evaluated XDAC-D, based on *KcBERT-Base* (Lee, 2020), against several baselines for LGC detection using the *MGTBench* framework (He et al., 2023). The baseline models are as follows:

Metric-based Detection Methods: We employed 12 methods: Log-Likelihood (Solaiman et al., 2019a), Rank, Log-Rank, Entropy, and GLTR (Gehrmann et al., 2019), Binoculars (Hans et al., 2024), LLMDeviation and MFD (Wu and Xiang, 2023), DetectGPT (Wang et al., 2023b), LRR and NPR (Su et al., 2023), FastDetectGPT (Bao et al., 2023).

LM-based Detection Methods: We evaluated four models for LGC detection: OpenAI-D (Solaiman et al., 2019a) and ChatGPT-D (Guo et al., 2023), pre-trained for detecting LLM-generated text and used without fine-tuning, and ConDA (Bhattacharjee et al., 2023) and LM-D (Ippolito et al., 2020), fine-tuned on the LGC dataset using *KcBERT*.

LLM-based Detection Methods: 1) Few-shot In-context Learning-based Detector: We leveraged *GPT-4o* (Achiam et al., 2023a)’s in-context learning capabilities for a few-shot detection model, operating without parameter updates. It was tested across 0-shot, 20-shot, and 100-shot settings using N_{HWC} and N_{LGC} examples. 2) Fine-tuning the GPT-4o: We fine-tuned *GPT-4o-2024-08-06*, one of the most advanced LLMs, using 400K comments due to service limitations.

4.2 LLM Attribution Models

For LLM attribution, we implemented and compared four models: *OpenAI-D*, *ChatGPT-D*, *LM-D* as our baseline, and our proposed models, XDAC-A, all fine-tuned on the LGC dataset using *KcBERT*. We developed two variants of XDAC-A: XDAC-AM for model-level attribution (identifying specific LLM models among 14 LLMs) and XDAC-AF for family-level attribution (classifying LLMs into 7 families such as GPT-series and Llama-series).

5 Evaluation

5.1 LGC Detection

We present experimental results and detailed analyses that demonstrate the effectiveness of our proposed model, *XDAC-D*, compared to various baseline models across different methodologies.

Comparison with Metric-based Models Table 3 compares the performance of XDAC-D with various metric-based models. Among the metric-based models, *MFD* achieved the highest F1 score (77.2%), but these models lack parameter updates and fail to capture LGC-specific characteristics, making them less effective. In contrast, XDAC, optimized for LGC detection, significantly outperformed all metric-based models. We evaluated the commercial LLM detection service GPTZero, using its API version that does not impose a minimum input length restriction. While effective for long English texts, it struggled with short comments, achieving an F1 score of 41.7%.

Model	Total	HWC	LGC	L-Text	M-Text	S-Text
Loglikelihood	77.1	83.2	72.7	76.8	79.7	64.6
Rank	63.8	6.8	88.1	66.5	73.2	33.3
LogRank	74.5	82.1	69.5	74.7	77.3	60.6
Entropy	66.0	71.0	62.7	68.2	69.4	47.5
GLTR	68.2	79.2	62.0	67.3	72.0	52.9
Binoculars	62.3	71.8	57.4	62.9	66.4	44.4
LLMDeviation	70.0	81.0	63.5	69.9	73.1	55.4
MFD	77.2	82.7	73.2	77.3	79.8	64.7
DetectGPT	62.0	22.4	78.0	64.0	70.5	33.6
DetectLLM-LLR	62.9	62.5	62.2	63.1	68.3	41.7
DetectLLM-NPR	31.0	77.6	22.2	32.2	32.9	20.8
FastDetectGPT	71.8	75.0	69.2	72.9	75.8	52.4
GPTZero	41.7	60.9	53.5	-	-	-
XDAC-D	**98.5**	**97.4**	**99.3**	**99.1**	**98.9**	**94.1**

Table 3: XDAC-D vs. Metric-based models.

Comparison with LM-based Models Table 4 compares XDAC-D with LM-based detection models. *OpenAI-D* and *ChatGPT-D* are English LLM detection models fine-tuned on RoBERTa-base, optimized for long-form text, resulting in limited performance on short Korean LGC. In contrast, models fine-tuned on KcBERT with LGC and HWC data, such as *ConDA* and *LM-D*, performed significantly better. *ConDA* achieved an F1 score of 94.9, while *LM-D* reached 95.3. Enhancing *LM-D* with Repetitive Pattern Transformation and XAI in XDAC-D led to a 68.1% relative improvement, where *relative↑* is defined as $(\text{new} - \text{old})/(100 - \text{old})$, quantifying the improvement relative to the theoretical maximum (Bao et al., 2023, Table 1). XDAC-D outperformed all models. While all LM-based models showed a decline in F1 score for short texts, XDAC-D maintained strong performance, scoring 99.1% for long texts and 94.1% for short texts, surpassing the baseline.

Comparison with LLM-based Models Table 5 compares XDAC-D with LLM-based detection models, including few-shot in-context learning and fine-tuned approaches. For few-shot models, N-shot indicates the number of reference comments provided. We evaluated 0-shot, 20-shot, and 100-shot configurations, observing minimal improvement beyond 20-shot (100-shot: F1 = 66.7%), which demonstrates the limitations of few-shot approaches for LGC detection.

Model	OpenAI-D	ChatGPT-D	ConDA	LM-D	XDAC-D
Total	52.2	67.8	94.9	95.3	**98.5 (relative↑ 68.1%)**
HWC	66.9	0.0	**98.9**	98.8	97.4
LGC	46.4	**99.9**	91.3	92.1	99.3
L-Text	53.5	70.7	94.8	95.1	**99.1**
M-Text	55.9	78.2	96.3	96.8	**98.9**
S-Text	35.2	34.8	93.9	93.5	**94.1**

Table 4: XDAC-D vs. LM-based models.

Training Method	Zeroshot Learning	20-Shots Learning	100-Shots Learning	GPT4o Finetuning	XDAC-D
# Data	0	20	100	400,000	2,000,000
Param Update	*X*	*X*	*X*	*O*	*O*
Inference Time (comments/sec)	14	13.5	10.3	0.2	**428.3**
Total	37.5	64.4	66.7	98.2	**98.5 (relative↑ 16.7%)**
HWC	98.6	**98.7**	96.9	97.7	97.4
LGC	23.4	48.0	51.5	98.7	**99.3**
L-Text	51.7	71.4	74.4	**99.3**	99.1
M-Text	34.3	64.1	66.7	**98.9**	**98.9**
S-Text	11.3	44.0	43.4	91.4	**94.1**

Table 5: Effects of LLM training approaches.

We also compared XDAC-D with a fine-tuned *GPT-4o* model. Despite being trained on a smaller data subset (due to resource constraints), the *GPT-4o* model achieved an F1 score of 98.2%, slightly lower than XDAC-D's 98.5%, yielding a 16.7% relative improvement. LLM-based models, despite their capabilities, process only 0.2 comments/s and incur a cost of $0.00023 per comment. XDAC-D, being a locally executable model, avoids these limitations, processing 428.3 comments/s at no per-comment expense. This local execution capability makes XDAC-D ideal for large-scale applications where resources are limited.

Ablation Study We conducted an ablation study to assess the contribution of key components, specifically *Linguistic Patterns* and *Repetitive Pattern Transformation*. As shown in Table 6, removing *Linguistic Patterns* reduced the F1 score to 98.0%, and excluding both dropped it to 95.3%. These results emphasize the importance of these modules, particularly *Repetitive Pattern Transformation*, which significantly boosts performance by handling repetitive patterns in the data.

Methods	F1 (Detection)
XDAC-D	**98.5**
w/o Linguistic Patterns	98.0
w/o Repetitive Patterns	96.3
w/o Linguistic Patterns and Repetitive Patterns	95.3

Table 6: Ablation studies of XDAC-D.

5.2 LLM Attribution

Table 7 presents the classification results. XDAC achieved F1 scores of 74.0% for model-level attribution and 84.3% for family-level attribution, effectively capturing LGC-specific linguistic features even on short text. Detection of *gemini-10-pro* and *claude-3-5-sonnet* was relatively easier, whereas *gpt-4o* posed the greatest challenge. Analysis of the confusion matrix (Appendix B.3) revealed that models within the same family were often misclassified due to stylistic similarities. Family-level attribution significantly reduced this confusion and improved overall classification performance.

Model	OpenAI-D	ChatGPT-D	LM-D	XDAC AM	XDAC AF
Total	52.5	26.6	52.2	**74.0**	**84.3**
L-Text	52.7	28.1	68.0	**75.2**	**86.8**
M-Text	51.6	27.7	66.2	**73.2**	**83.8**
S-Text	44.4	26.6	52.2	**71.1**	**76.4**
GPT Family					**86.7**
GPT-3.5	54.4	14.3	55.8	**79.9**	-
GPT4-pre	57.7	0.0	0.0	**77.3**	-
GPT4o-05	32.5	27.1	40.0	**59.6**	-
GPT4o-08	36.6	10.8	16.7	**59.7**	-
Claude Family					**81.6**
3.5-sonnet	59.3	41.7	45.5	**81.0**	-
Gemini Family					**82.2**
1.0-pro	52.4	0.0	50.0	**81.9**	-
HCX Family					**88.1**
HCX-1	58.2	59.8	73.1	**76.0**	-
HCX-3	55.2	48.1	69.3	**74.8**	-
Gemma Family					**86.0**
Gemma-9B	54.4	40.8	66.7	**72.8**	-
Gemma-27B	50.1	18.2	57.4	**72.0**	-
Qwen Family					**81.4**
Qwen-7B	55.9	35.9	73.1	**74.1**	-
Qwen-32B	46.5	16.4	65.5	**72.0**	-
Llama Family					**84.1**
Bllossom-8B	60.6	27.3	52.5	**78.4**	-
Bllossom-70B	60.5	32.7	64.7	**77.0**	-

Table 7: Model-level (AM) and family-level (AF) F1 scores for LGC attribution.

5.3 Real-World Deployment and Analysis

We demonstrated XDAC's practicality by analyzing 5.2M comments from a major Korean news platform, posted since 2023, when LLM accessibility surged following OpenAI's release of ChatGPT-3.5 (OpenAI, 2022) in late 2022. The analysis took 3.5 hours, identifying 108,132 (2.1%) as potential LGC. Among the top 25% high-probability cases, 27,029 were identified as likely LGC. An XAI-based analysis further revealed that their linguistic and repetitive patterns closely aligned with those found in known LGC. User ID grouping revealed accounts suspected of extensive LLM usage, demonstrating XDAC's effectiveness in large-scale LGC detection and establishing a foundation for identifying LLM-driven comment manipulation. We reported these findings to Naver, who acknowledged the significance of our results and the potential impact of our detection system for maintaining comment section integrity.

5.4 Robustness Against Adversarial Strategies

To assess XDAC's resilience against evasion attempts, we evaluated two adversarial strategies: (1) adding repeated human-like characters and (2) inserting human-characteristic patterns (Appendix F.3). Experimental results indicate that XDAC maintained over 96% accuracy with 20 repeated characters, whereas the baseline dropped to 84.7%. For human-characteristic pattern insertion, both models showed performance degradation, but XDAC consistently outperformed the baseline, which declined to 7.9% at 20% insertion and nearly 0% beyond 50%. When punctuation and colloquial markers such as "ㅋ" and "." were inserted within a comment (e.g., "좋은 기.사ㅋ 감사합?니다!"), the baseline struggled to distinguish generated text from HWC, while XDAC retained a measurable detection advantage, demonstrating improved robustness against adversarial manipulation.

6 Conclusion

This work introduces XDAC, an XAI-driven framework designed to address the challenge of detecting and attributing LGC in online news platforms. Through comprehensive linguistic pattern analysis, XDAC achieves state-of-the-art performance in both detection and attribution of short-form comments, advancing online content integrity.

Acknowledgments

Hyoungshick Kim (second author) was supported by IITP grant (No. RS-2024-00439762) and contributed primarily to the writing, analysis, and overall structure of the paper. Alice Oh (third author) was supported by IITP grant (No. RS-2024-00509258 and No. RS-2024-00469482) and contributed to the writing of the paper.

Limitations

Dataset. The primary limitation of our dataset is that it consists of LLM-generated news-related comments created by us using various LLM models rather than actual LLM-generated content posted online. This approach was necessitated by the inherent unavailability of verifiable ground-truth labels for real-world LLM usage. Following standard practice in machine-generated text detection studies (He et al., 2023; Bao et al., 2023), we applied LLM models to real news articles, utilizing carefully designed prompts to elicit natural and contextually appropriate comments under controlled scenarios, thereby enhancing the realism and generalizability of our dataset. While we tried to generate human-like text, real-world scenarios might involve more advanced techniques and incorporate human feedback, potentially resulting in more sophisticated content.

To assess the generalizability of our approach, we conducted additional experiments on English-language social media comments from X, YouTube, and Instagram (400K comments). Our English version of XDAC achieved an F1 score of 97.6%, outperforming baseline methods with a 35.1% relative improvement over LM-D. The model also demonstrated strong attribution capabilities, achieving F1 scores of 69.1% for model-level and 82.8% for family-level attribution tasks. However, while these preliminary English results are promising, they are based on a relatively limited evaluation set and would require more extensive validation. Unlike our Korean dataset, which underwent comprehensive testing across various scenarios and attack methods, the English and other language applications of XDAC require more thorough verification with larger-scale data and diverse evaluation settings. Additionally, despite our efforts to represent diverse topics and writing styles, the characteristics of news comments may differ significantly from other types of online discourse, suggesting the need for more comprehensive cross-domain validation.

Experiments. Our experiments focused on developing and evaluating our XDAC model for detecting LGC. We did not exhaustively optimize hyperparameters or conduct extensive ablation studies, which might yield better performance. Additionally, our real-world analysis was conducted over a specific time frame (Jan. 2023 to Aug. 2024), which may not capture the full spectrum of LLM advancements and their impacts on online discourse.

Ethical Considerations

Mitigating Malicious Use. We acknowledge the potential for our research to be misused to generate harmful content. However, we believe that openly discussing these vulnerabilities offers more benefits than risks. Our approach encourages the broader community to consider adaptive adversaries when developing countermeasures. To minimize potential abuse, we have implemented several safeguards: we limit the release of our comment generation process details, only share the detection model code publicly, and incorporate content moderation filters in our system. We are actively collaborating with platform moderators to integrate our detection system and are committed to ongoing research in adversarial robustness. Additionally, we are reaching out to relevant stakeholders to enhance the resilience of existing tools. We plan to open-source our framework and findings upon acceptance, following responsible disclosure practices. These measures aim to balance the benefits of our research with responsible AI development and deployment.

Data Privacy. To ensure data privacy in our real-world analysis, we collected human-written comments from publicly available news platforms using their official APIs rather than scraping news data. All user information in our dataset is anonymized by replacing identifiable information with alphanumeric IDs to protect individual privacy.

References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023b. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Advanced information networking and applications: Proceedings of the*

34th international conference on advanced information networking and applications (AINA-2020), pages 1341–1354. Springer.

Alim Al Ayub Ahmed, Ayman Aljarbouh, Praveen Kumar Donepudi, and Myung Suh Choi. 2021. Detecting fake news using machine learning: A systematic literature review. *Psychology And Education*, 58(1).

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. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Ben Buchanan, Andrew Lohn, Micah Musser, and Katerina Sedova. 2021. Truth, lies, and automation. *Center for Security and Emerging technology*, 1(1):2.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*.

ChangSu Choi, Yongbin Jeong, Seoyoon Park, InHo Won, HyeonSeok Lim, SangMin Kim, Yejee Kang, Chanhyuk Yoon, Jaewan Park, Yiseul Lee, et al. 2024. Optimizing language augmentation for multilingual large language models: A case study on korean. *arXiv preprint arXiv:2403.10882*.

Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. 2023. Chatgpt and the rise of large language models: the new ai-driven infodemic threat in public health. *Frontiers in public health*, 11:1166120.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021a. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021b. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Shangbin Feng, Herun Wan, Ningnan Wang, Zhaoxuan Tan, Minnan Luo, and Yulia Tsvetkov. 2024. What does the bot say? opportunities and risks of large language models in social media bot detection. *arXiv preprint arXiv:2402.00371*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Margherita Gambini, Marco Avvenuti, Fabrizio Falchi, Maurizio Tesconi, and Tiziano Fagni. 2023. Detecting generated text and attributing language model source with fine-tuned models and semantic understanding. In *IberLEF@ SEPLN*.

Henrique Da Silva Gameiro, Andrei Kucharavy, and Ljiljana Dolamic. 2024. Llm detectors still fall short of real world: Case of llm-generated short news-like posts. *arXiv preprint arXiv:2409.03291*.

Sebastian Gehrmann et al. 2019. Gltr: Statistical detection and visualization of generated text. In *ACL: System Demonstrations*, pages 111–116.

Wooyoung Go and Daewoo Lee. 2018. Toward trustworthy deep learning in security. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 2219–2221, New York, NY, USA. Association for Computing Machinery.

Josh A Goldstein, Jason Chao, Shelby Grossman, Alex Stamos, and Michael Tomz. 2024. How persuasive is ai-generated propaganda? *PNAS nexus*, 3(2).

Josh A Goldstein, Girish Sastry, Micah Musser, Renee DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*.

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. *arXiv preprint arXiv:2301.07597*.

Zhen Guo and Shangdi Yu. 2023. Authentigpt: Detecting machine-generated text via black-box language models denoising. *arXiv preprint arXiv:2311.07700*.

Philipp Hacker, Andreas Engel, and Marco Mauer. 2023. Regulating chatgpt and other large generative ai models. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1112–1123.

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*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *arXiv preprint arXiv:2303.14822*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Shan Jiang and Christo Wilson. 2018. Linguistic signals under misinformation and fact-checking: Evidence from user comments on social media. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–23.

Ji Won Kim and Gina Masullo Chen. 2021. Exploring the influence of comment tone and content in response to misinformation in social media news. *Journalism Practice*, 15(4):456–470.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Tharindu Kumarage, Amrita Bhattacharjee, Djordje Padejski, Kristy Roschke, Dan Gillmor, Scott Ruston, Huan Liu, and Joshua Garland. 2023a. J-guard: Journalism guided adversarially robust detection of ai-generated news. *arXiv preprint arXiv:2309.03164*.

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

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

Kavita Kumari, Alessandro Pegoraro, Hossein Fereidooni, and Ahmad-Reza Sadeghi. 2023. Demasq: Unmasking the chatgpt wordsmith. *arXiv preprint arXiv:2311.05019*.

Junbum Lee. 2020. Kcbert: Korean comments bert. In *Annual Conference on Human and Language Technology*, pages 437–440. Human and Language Technology.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt. *arXiv preprint arXiv:2306.05524*.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2024. On the detectability of chatgpt content: benchmarking, methodology, and evaluation through the lens of academic writing. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2236–2250.

Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting fire with fire: The dual role of llms in crafting and detecting elusive disinformation. *arXiv preprint arXiv:2310.15515*.

Luca Luceri, Eric Boniardi, and Emilio Ferrara. 2024. Leveraging large language models to detect influence campaigns on social media. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1459–1467.

George K Mikros, Athanasios Koursaris, Dimitrios Bilianos, and George Markopoulos. 2023. Ai-writing detection using an ensemble of transformers and stylometric features. In *IberLEF@ SEPLN*.

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

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

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE symposium on security and privacy (SP)*, pages 1613–1630. IEEE.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Kai Shu, Suhang Wang, Thai Le, Dongwon Lee, and Huan Liu. 2018. Deep headline generation for clickbait detection. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 467–476. IEEE.

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

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

Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. 2023. Ai model gpt-3 (dis) informs us better than humans. *Science Advances*, 9(26):eadh1850.

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*.

Anthropic Team. 2024. Claude 3.5 sonnet model card addendum. *https://www-cdn.anthropic.com/*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

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

Herun Wan, Minnan Luo, Zihan Ma, Guang Dai, and Xiang Zhao. 2024. How do social bots participate in misinformation spread? a comprehensive dataset and analysis. *arXiv preprint arXiv:2408.09613*.

Hong Wang, Xuan Luo, Weizhi Wang, and Xifeng Yan. 2023a. Bot or human? detecting chatgpt imposters with a single question. *arXiv preprint arXiv:2305.06424*.

Rongsheng Wang, Qi Li, and Sihong Xie. 2023b. Detectgpt-sc: Improving detection of text generated by large language models through self-consistency with masked predictions. *arXiv preprint arXiv:2310.14479*.

Zhendong Wu and Hui Xiang. 2023. Mfd: Multi-feature detection of llm-generated text. *Research Square preprint*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, Hyunwook Kim, Kyung-Min Kim, Munhyong Kim, Sungju Kim, et al. 2024. Hyperclova x technical report. *arXiv preprint arXiv:2404.01954*.

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

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2020. Defending against neural fake news. *Neurips*.

Thomas Zerback and Florian Töpfl. 2022. Forged examples as disinformation: The biasing effects of political astroturfing comments on public opinion perceptions and how to prevent them. *Political Psychology*, 43(3):399–418.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Wanjun Zhong, Duyu Tang, Zenan Xu, Ruize Wang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Neural deepfake detection with factual structure of text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2461–2470.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Dataset Statistics

As Table 8 shows, while mean word counts for HWC and LGC are similar, LGC exhibit significantly smaller standard deviation and IQR. This indicates LGC have more consistent and less varied lengths than HWC, which show wider distribution, highlighting distinct characteristics between human and LLM-generated content.

	HWC		LGC	
	char-level	**word-level**	**char-level**	**word-level**
mean $\pm$ sd	51 ± 51	11 ± 11	44 ± 18	11 ± 4
median (IQR)	35 (45)	8 (10)	42 (23)	10 (5)

Table 8: Length Statistics of HWC and LGC.

B LGC Generation Detail

B.1 LLM Model Selection

Selecting appropriate LLMs is crucial as they directly influence the linguistic style, complexity, and emotional tone of the generated comments. In our framework, we experimented with various state-of-the-art LLM models for the Korean LGC generation. We considered both API-based and open-source models based on hardware requirements and performance capabilities. For instance, an NVIDIA A100 80G GPU can handle models with up to 27B parameters, but not 80B models. Table 9 presents the LLM models used for Korean LGC generation in our experiments, along with their availability and primary language. We selected models that are suitable for generating Korean LGC. Specifically, we included HCX (Yoo et al., 2024), a model exclusively trained for Korean. Additionally, the Bllossom (Choi et al., 2024) model was chosen as an open-source Korean model, as it was fine-tuned on LLaMA 3 (Touvron et al., 2023) with a focus on the Korean language. While the primary language of OpenAI's GPT series (Achiam et al., 2023b) (GPT-3.5, GPT-4, GPT-4o), Google's Gemini (Reid et al., 2024) and Gemma-2 (Team et al., 2024) (Gemma-2-9B, Gemma-2-27B), Anthropic's Claude 3.5 sonnet (Team, 2024), and Alibaba Cloud's Qwen2.5 (Qwen2.5-7B, Qwen2.5-32B) (Yang et al., 2024) is either English or Chinese, they were included because they also perform well in Korean text generation.

For clarity, we use the following abbreviations: **GPT-3.5** (gpt-3.5-turbo-0125), **GPT4-pre** (gpt-4-0125-preview), **GPT4o-05** (gpt-4o-2024-05-13), **GPT4o-08** (gpt-4o-2024-08-06), **Claude** (claude-3.5-sonnet-20240620), **Gemini** (gemini-1.0-pro), **HCX-1** (HCX-DASH-001), **HCX-3** (HCX-003), **Gemma-9B** (gemma-2-9b-it), **Gemma-27B** (gemma-2-27b-it), **Qwen-7B** (Qwen2.5-7B-Instruct), **Qwen-32B** (Qwen2.5-32B-Instruct), **Bllossom-8B** (llama-3-Korean-Bllossom-8B), **Bllossom-70B** (llama-3-Korean-Bllossom-70B).

B.2 Enhancing Comment Naturalness

To make generated comments closely resemble genuine user input. We employed strategies that mimic human writing characteristics. These strategies, outlined in Table 10, are incorporated into the prompt.

These techniques collectively contribute to creating more realistic comments that closely resemble human writing, effectively improving the naturalness of generated content and potentially enhancing its relevance and engagement.

B.3 Sentiment Subtype Selection

LGC generation framework selects specific sentiment subtypes and integrates them into the prompt to generate more diverse and realistic comments. This process is crucial in shaping the tone and reflecting the intended sentiment of news discussions. We define 32 positive and 37 negative sentiment subtypes, including the "None" subtype, outlined in Tables 11 and 12. These subtypes were meticulously derived from an analysis of sentiment expressions in HWC, as existing taxonomies typically offer only broad classifications (e.g., positive, negative, neutral) that lack the granularity for real-world news discussions. This approach enhances the diversity and realism of generated LGC, accurately capturing varied perspectives.

B.4 Reference-Augmented Generation Strategies

LGC generation framework presents four distinct comment generation approaches, each with unique advantages. *Generation without reference* produces creative responses without external context, risking relevance or coherence. *News-based generation* ensures topical relevance by referencing articles but may lack the nuances of user comments. *News and comment-based generation* integrates both articles and existing comments, enhancing engagement but potentially reinforcing biases. *Opinion-based generation* aligns comments with predefined viewpoints,

Provider	Availability	Primary Language	Model
OpenAI (API)	Private	English	gpt-3.5-turbo-0125 gpt-4-0125-preview gpt-4o-2024-05-13 gpt-4o-2024-08-06
Google (API)	Private	English	gemini-10-pro
Anthropic (API)	Private	English	claude-3-5-sonnet-20240620
Naver (API)	Private	Korean	HCX-DASH-001 HCX-003
Google	Public	English	gemma-2-27b-it gemma-2-9b-it
META (Bllossom)	Public	Korean	llama-3-Korean-Bllossom-8B llama-3-Korean-Bllossom-70B
Alibaba Cloud	Public	Chinese	Qwen2.5-7B-Instruct Qwen2.5-32B-Instruct

Table 9: LLM models and their availability and primary language for Korean LGC generation.

ID	Comment Strategies
S1	Use informal, conversational language.
S2	Include emojis throughout the text.
S3	Introduce minor textual variations.
S4	Express emotions naturally.
S5	Employ special characters for emphasis.
S6	Keep comments concise.

Table 10: Strategies for enhancing comment naturalness.

enabling targeted messaging but risking bias and polarization.

B.5 Prompt Template for LLM-Based Comment Generation

As shown in Figure 4, the prompt template is designed for generating fake comments that appear similar to real user comments on news articles. The instructions guide the creation of AI-generated comments to ensure they blend in naturally, specifying details such as the number of comments, sentiment, and language. The template also emphasizes that the generated comments should be unique and in line with the style of actual news comments. The output must follow a structured list of strings to facilitate integration into other systems. In this prompt template, the red, bolded text within curly braces (e.g., **{Sentiment_Category}**) represents user-provided input. Each placeholder corresponds to a specific parameter required for generation:

- **{Sentiment_Category}**: Specifies the overall sentiment to be reflected in the generated comments (e.g., Positive, Negative).
- **{Sentiment_Subtype_Selection}**: Further refines the sentiment by selecting a detailed subtype. Available subtypes are listed in Tables 11 and 12.
- **{Number_of_Comments}**: Specifies the number of comments to be generated in a single LGC generation process. The default value is 10.
- **{Comment_Language}**: Sets the language for comment generation (e.g., Korean, English, French). Any language supported by the selected LLM model can be used.
- **{Enhancing_Comment_Naturalness}**: Additional strategies to improve the naturalness of generated comments, such as using informal expressions, emojis, minor variations, or special characters. See Table 10 for detailed strategies.
- **{Reference_News}**: The news article content used as the generation source when applying Reference-Augmented Generation strategies.
- **{Reference_News_Comments}**: Real user comments provided as stylistic references. This input is used in comment-based or news-comment hybrid Reference-Augmented Generation, which can enhance engagement but may also reinforce pre-existing biases.

C Model-Based Evaluation Details for LGC

Our test dataset comprises 5,600 comments generated by 14 LLM models, covering various sentiment categories, temperature settings, and reference types. The evaluation methodology integrates three key components into a single prompt: a quality assessment based on four criteria (fluency, specificity,

Types	Subtypes
None Type	None
Positive Evaluation	Content Evaluation: Positive response and praise for the content
	Approval Opinion: Agreement and support for the content of the article
	Informative: Emphasis on the richness and helpfulness of the information
	Fun: Highlighting the fun and interesting elements of the article
Emotional Response	Joy and Happiness: Expression of positive emotional reactions
	Moved and Hopeful: Emphasis on touching elements and positive outlook
	Gratitude and Respect: Words of gratitude and respect for the article and its author
Support and Empathy	Topic Support: Strong expression of support for the topic of the article
	Expression of Empathy: Highlighting empathy and solidarity among readers
	Praise for Problem Solving: Recognition of contributions to social problem solving
Information Reliability and Truthfulness Positive Evaluation	Source Reliability: Emphasizing the reliability of information sources
	Praise for Evidence: Evaluation and praise for the presentation of evidence
	Information Reliability: Positive evaluation of overall information reliability
Encouragement and Praise	Encouragement for Author: Encouragement for the article's author
	Institution Evaluation: Positive evaluation of related institutions or media outlets
	Support for Activities: Support for reporting and news coverage activities
Additional Information Provided	Additional Information: Providing additional information related to the article content
	Sharing Experience: Sharing related experiences and knowledge
	Presenting Different Perspectives: Presenting different viewpoints on the article content
Constructive Discussion and Opinion Offering	Participation in Discussion: Constructive discussion on the article content
	Exchange of Opinions: Presentation and exchange of diverse opinions
	Advancing Discussion: Advancing discussion in a positive direction
Social Impact and Value Expectation	Expectation of Change: Expectations for social change
	Value Praise: Praising values that contribute to societal development
	Positive Expectations for the Future: Positive expectations for a bright future
Humor and Positive Emotion Expression	Use of Humor: Expression of humor and jest
	Expression of Emotions: Various expressions of positive emotions
	Creating Atmosphere: Creating a bright and warm atmosphere
Recommendation and Endorsement	Dissemination of Content: Recommendation and dissemination of article content
	Expression of Recommendation: Recommending the article to others
	Gratitude for Information: Gratitude for the provision of good information

Table 11: Positive sentiment subtypes.

Types	Subtypes
None Type	None
Critical Analysis	Criticism of the Article's Structure/Logical Completeness
	Criticism of the Reporting Style/Perspective
	Criticism of the Accuracy/Fairness of the Content
Emotional Reaction	Anger/Rage/Fury
	Sadness/Mourning/Grief
	Anxiety/Worry/Fear
Ethical Criticism	Pointing Out Violations of Journalism Ethics
	Criticism of Morality and Values
	Criticism of Social Impact and Consequences
Doubts about Information Reliability and Truthfulness	Suspicions about Source/Data Reliability
	Pointing Out Lack of Evidence
	Raising Concerns about Errors/Distortions
	Questioning the Facts
	Allegations of Intentional Distortion
	Distrust of the Article
Personal Attacks and Insults	Defamation/Personal Attack on the Author
	Derogatory Remarks Towards an Individual
Correction Request	Pointing Out Errors and Demanding Corrections
	Asking for Additional Information Updates
Promotion of Social Division	Discrimination/Blame of Specific Groups
	Expression of Hatred
	Incendiary Statements and Escalation of Conflict
Defensive Behavior	Claiming Victimhood for Oneself or One's Group
	Defending a Specific Ideology/Perspective
Passive Dissatisfaction	Complaints and Grievances Without Constructive Alternatives
Suggestions and Solutions	Pointing Out Problems and Proposing Directions for Improvement
	Offering Specific Solutions/Alternatives
Warnings and Concerns	Alerting to Social Risks
	Expressing Concerns about Environmental Threats
	Economic Implications Concern
Comparison with Other Figures/Events	Similarities with Other Figures/Events
	Differences with Other Figures/Events
Level of Empathy	Underestimating the Issue
	Light Reaction
Humor/Satire	Humor/Jesting/Jokes
	Satire/Sarcasm/Ridicule

Table 12: Negative sentiment subtypes.

```
## Instructions:
1. Generate comments on the news.
2. Ensure that AI-generated comments do not
   stand out and appear natural as if written
    by a human.
3. Please create {Sentiment_Category}
   comments {Sentiment_Subtype_Selection}
4. Provide {Number_of_Comments} number of
   comments
5. Please generate the comments in
   {Comment_Language}
6. When making multiple requests, each
   generated result should be unique, as if
   written by a human.
{Enhancing_Comment_Naturalness}
8. Please make comments with reference to the
    text below.
{Reference_News}
9. Based on the actual comments below, please
    create comments in a similar style.
{Reference_News_Comments}
10. The format of the generated comments
   should follow the structure below and must
    be in the form of a list of strings.
## Output format: list
["comment1", "comment2", "comment3", ...]
```

Figure 4: Prompt template used in the LGC generation framework.

coherence, and fairness), an evaluation of comment alignment with the author's intent on the prompt (content relevance, comment authenticity reflection, sentiment subtype reflection), and an assessment of sentiment accuracy (positive as positive, negative as negative).

For evaluation, GPT-4o was employed as an LLM-as-a-Judge, following prior work (Choi et al., 2024). For each LGC, GPT-4o was prompted to make a binary judgment (True/False) on whether the comment satisfied each evaluation criterion. The final score for each category, as presented in Table 1, was then computed as the proportion of LGC that received a "True" judgment. For each LGC, GPT-4o was prompted to make a binary judgment (True/False) on whether the comment satisfied each evaluation criterion. The final score for each category, as presented in Table 1, was then computed as the proportion of LGC that received a "True" judgment. This automated evaluation process allows for consistent and scalable assessment of the large volume of generated comments. This comprehensive approach enables a thorough analysis of LGC, offering insights into its authenticity, relevance, and alignment with the intended sentiments and authorial intent.

The LGC Evaluation prompt is provided in Figure 5. In this prompt template, the red, bolded text within curly braces (e.g., **{Reference_News}**) represents user-provided input. Each placeholder corresponds to a specific input required for evaluating LLM-generated comments:

- **{Reference_News}**: This field should be filled only when the news article was used during LGC generation, such as in the *News-based*, *News-and-Comment-based*, or *Opinion-based* generation strategies.
- **{Reference_News_Comments}**: This field should be filled only when real user comments were used during LGC generation, such as in the *News-and-Comment-based* or *Opinion-based* generation strategies.
- **{Sentiment_Subtype_Selection}**: This field should always be filled to indicate the intended sentiment subtype that the generated comment is expected to express.
- **{Enhancing_Comment_Naturalness}**: This field should always be filled to describe the naturalness-enhancing strategies applied during generation.

C.1 Sentiment Reflection Evaluation

Figures 6 and 7 present the evaluation results for sentiment reflection in Korean LGC, divided into positive and negative sentiment prompts, respectively.

Figure 6 shows that most models demonstrate strong performance in reflecting positive sentiment. Notably, *Qwen2.5-7B-Instruct* achieves the highest accuracy. However, several models, such as *GPT-4o-2024-08-06*, exhibit a tendency to neutralize positive prompts, often generating more moderate expressions than intended.

In contrast, Figure 7 illustrates that generating negative sentiment LGC is generally more challenging across all models. The accuracy of reflecting negative sentiment is consistently lower than for positive sentiment. Models such as *Claude-3.5-Sonnet-20240620* and *HCX-DASH-001* frequently convert negative prompts into neutral expressions, suggesting a bias toward moderation or reluctance to produce strongly negative content.

These observations reveal a clear asymmetry in sentiment handling capabilities among models. While positive sentiment is more reliably preserved, maintaining the intended tone in negative sentiment

```
### Task Description:
Determine whether the news comment was
   generated by an LLM and analyze it.
Please do not generate any other opening,
   closing, or explanations. Respond in JSON
   format with the following items:

1) specificity: Evaluate whether the comment
   provides specific and clear information.
   Answer with "True" or "False".
2) reason_of_specificity: Explain why you
   classified 'specificity' as either "True"
   or "False".
3) fairness: Determine if the comment is fair
    and unbiased, providing a balanced view.
   Answer with "True" or "False".
4) reason_of_fairness: Explain why you
   classified 'fairness' as either "True" or
   "False".
5) relevance: Determine whether the entered
   news comment is relevant to the news text.
    Answer with "True" or "False".
6) reason_of_relevance: Explain why you
   classified 'relevance' as either "True" as
    "False".
7) reflect_sentiment_type: Determine if the
   comment accurately reflects the sentiment
   type below. Answer with "True" or "False".
8) reason_of_reflect_sentiment_type: Explain
   why you classified 'reflect_sentiment_type
   ' as either "True" or "False".
9) reflect_comment_authenticity: Determine if
    the comment accurately reflects the
   sentiment type below. Answer with "True"
   or "False".
10) reason_of_reflect_comment_authenticity:
   Explain why you classified '
   reflect_comment_authenticity' as either "
   True" or "False".

### The news context
{Reference_News}

### The news comment
{Reference_News_Comments}

### The sentiment type
{Sentiment_Subtype_Selection}

### The comment authenticity
{Enhancing_Comment_Naturalness}

### Answer in JSON format:
```

Figure 5: Prompt template for model-based LGC evaluation.

prompts remains a significant challenge in short-form LGC.

D Human Evaluation Details for LGC

D.1 Evaluation Setup

Dataset We sampled 140 LGC and 70 HWC, resulting in 210 comment instances. Each comment was paired with a corresponding news article, which included a headline and a truncated body (up to 400 characters).

Evaluators Four adult evaluators (aged 30–49), all holding undergraduate degrees, participated in the study. Two had professional experience in the IT industry, while the others had academic backgrounds in psychology and economics.

Evaluation Aspects Evaluators assessed each comment along six aspects using a 3-point Likert scale:

1. **LLM Authorship**: Whether the comment was perceived as human- or LLM-generated (options: Human-written, Uncertain, LLM-generated).
2. **Relevance**: How well the comment relates to the news article (options: Relevant, Unclear, Irrelevant).
3. **Specificity**: Whether the comment contains concrete and informative content (options: Specific, Unclear, Not specific).
4. **Fairness**: Whether the comment expresses a balanced and unbiased viewpoint (options: Fair, Unclear, Biased).
5. **Fluency and Naturalness**: Whether the comment is grammatically correct and natural (options: Yes, Unclear, No).
6. **Sentiment**: The sentiment expressed in the comment (options: Positive, Neutral, Negative).

Procedure Evaluators were presented with each article–comment pair in randomized order, with no indication of whether the comment was LLM- or human-generated. All 210 comments were independently rated by each evaluator across the six aspects, using the questionnaire format shown in Figure 8. The average rating for each aspect was computed across all four evaluators.

D.2 Human Evaluation Results

Table 13 presents the detailed human evaluation results, comparing LGC and HWC across six distinct quality dimensions. We analyze these findings in the following paragraphs.

LLM Authorship Annotators judged 67.1% of LGC as human-written, compared to 72.9% of HWC. This suggests that many LGC are perceived as indistinguishable from human-authored text.

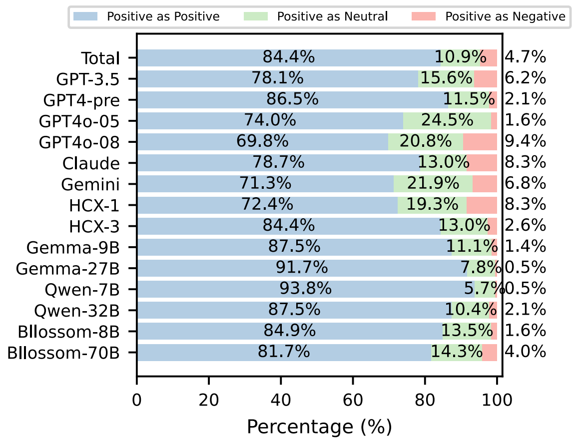


Figure 6: Positive sentiment reflection evaluation for LGC generation.

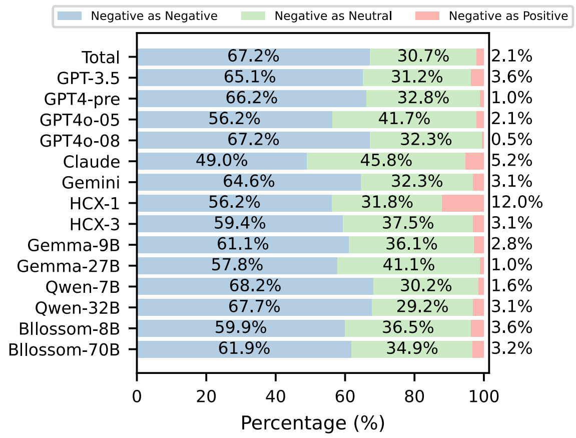


Figure 7: Negative sentiment reflection evaluation for LGC generation.

Evaluation Aspect	Type	Option 1	Option 2	Option 3
LLM Authorship	Option	Human-written	Uncertain	LLM-generated
	LGC	67.1%	13.6%	19.3%
	HWC	72.9%	16.7%	10.5%
Relevance	Option	Relevant	Unclear	Irrelevant
	LGC	94.8%	2.6%	2.6%
	HWC	87.1%	1.4%	11.4%
Specificity	Option	Specific	Unclear	Not specific
	LGC	49.5%	31.3%	19.3%
	HWC	41.8%	29.3%	28.9%
Fairness	Option	Fair	Unclear	Biased
	LGC	34.8%	32.0%	33.2%
	HWC	23.2%	26.8%	50.0%
Fluency & Naturalness	Option	Yes	Unclear	No
	LGC	71.3%	23.0%	5.7%
	HWC	44.6%	18.6%	36.8%
Sentiment	Option	Positive	Neutral	Negative
	LGC	48.2%	13.9%	37.9%
	HWC	10.0%	12.1%	77.9%

Table 13: Human evaluation results comparing LGC and HWC across six quality aspects.

Relevance LGC were rated as relevant to the corresponding article in 94.8% of cases, outperforming HWC at 87.1%. This indicates that, with well-designed prompts, LLMs can generate highly contextually aligned comments.

Specificity LGC were rated as specific in 49.5% of cases, slightly higher than HWC at 41.8%. However, both comment types received a substantial proportion of "unclear" ratings—31.3% for LGC and 29.3% for HWC—highlighting the inherent ambiguity of short-form user comments.

Fairness Only 34.8% of LGC and 23.2% of HWC were judged to be fair. A notable portion of comments were rated as unclear in fairness (32.0% for LGC and 26.8% for HWC), while HWC were more frequently perceived as biased (50.0% vs. 33.2%), suggesting that LLMs may mitigate certain biases commonly found in user-generated content.

Fluency and Naturalness LGC showed substantially higher fluency, with 71.3% rated as grammatically correct and natural, compared to 44.6% for HWC. This demonstrates LLMs' strength in producing well-formed sentences even under constrained settings.

Sentiment LGC exhibited a more balanced sentiment distribution, with 48.2% positive and 37.9% negative sentiment. In contrast, HWC were overwhelmingly negative (77.9%) and only 10.0% were positive, consistent with common sentiment trends in real-world online comment sections.

E Linguistic Patterns Extraction for LGC

E.1 Training Detail for XAI-Driven Analysis

We trained a simple binary classification model based on 1D-CNN and KcBERT-base (109M). The model was developed using the PyTorch 2.0 framework and trained on an NVIDIA A100 GPU provided by Google Colab Pro. All experiments were conducted with 10 epochs of repeated training, and each experiment was completed within 24 hours. We used a learning rate of 1e-4 and a batch size of 256, with early stopping and threshold adjusting applied. A checkpoint was saved at the best epoch in terms of accuracy. The data was constructed using the LGC framework, and the training and test datasets were completely separated.

```
The following is an excerpt from a real news
   article. Please read it and answer the
   questions regarding the comment below.

### News Title:
{news_title}

### News Content:
{news_content}

### Comment:
{news_comments}

### Human Evaluation Questions:
Q1) Is this comment written by a human?
[1. Human-written, 2. Uncertain, 3. AI-
   generated]

Q2) Is this comment related to the news
   content?
[1. Related, 2. Uncertain, 3. Unrelated]

Q3) Does the comment provide specific and
   clear information?
[1. Specific, 2. Uncertain, 3. Not specific]

Q4) Is the comment fair and unbiased?
[1. Fair, 2. Uncertain, 3. Biased]

Q5) Is the comment grammatically correct and
   natural?
[1. Yes, 2. Uncertain, 3. No]

Q6) What is the sentiment of the comment?
[1. Positive, 2. Neutral, 3. Negative]
```

Figure 8: Example questionnaire presented to human evaluators for assessing LGC and HWC comments across six evaluation aspects.

E.2 XAI-Based Identification of Key Linguistic Features

To analyze distinguishing linguistic patterns between LGC and HWC, we applied XAI techniques to identify key features contributing to classification decisions. Figure 9 presents the frequency distribution of high-impact linguistic features extracted using a BERT-based model.

The results indicate that LGC exhibits a strong preference for formal and structured expressions such as "것 같다" ("it seems") and "에 대한" ("about"), which frequently appear in LGC. In contrast, HWC demonstrates greater variation, including informal expressions, repeated characters, and conversational markers, such as "ㅋㅋㅋㅋ" ("LOL") and "정말" ("really").

This analysis highlights systematic stylistic differences between human and AI-generated comments, underscoring the necessity of leveraging these distinct linguistic traits for LGC detection.

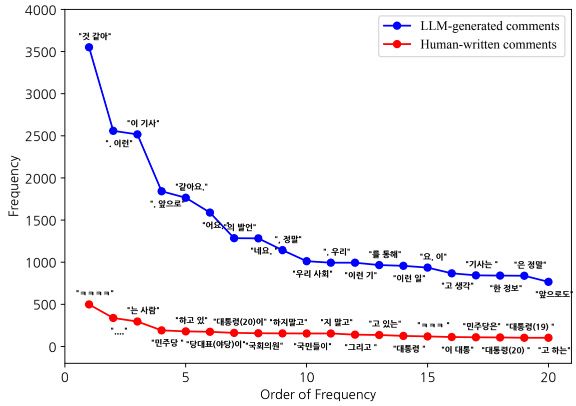


Figure 9: XAI-based identification of key linguistic features using BERT.

Algorithm 1 Repetitive pattern transformation

Input: Text sequence $\vec{x}$ with length s
Output: Transformed text with repetitive patterns encoded using <REP> and </REP> tokens
for each character c_i in $\vec{x}$ **do**
  Initialize $n = 1$
  while next character $c_{i+1} = c_i$ **do**
    Increment n ▷ Count repetition
    Move to the next character $i = i + 1$
  end while
  if $n > 1$ **then**
    Replace repeated sequence with <REP> c_i n </REP>
  else
    Keep c_i unchanged
  end if
end for
Return: Transformed text with repetitive patterns encoded

E.3 Repetitive Pattern Transformation

Algorithm 1 describes the repetitive pattern transformation process used to encode repeated characters and formatting symbols in text. This transformation introduces the special tokens "<REP>" and "</REP>" to explicitly mark repetitions, ensuring that stylistic features are preserved.

The algorithm processes each character in the input text and identifies consecutive repetitions. If a sequence of repeated characters is detected, it is replaced with a structured encoding, such as "<REP> ㅋ 5 </REP>" for "ㅋㅋㅋㅋㅋ". This encoding is also applied to spaces and line breaks, e.g., "<REP> <SPACE> 5 </REP>" for consecutive spaces and "<REP> <ENTER> 2 </REP>" for multiple line breaks.

Label	Predict	Example
HWC	HWC	지독하게 털어대는구나...털어서 안나오는것도 대단하다...
		ㅈㄹ하네 ㅋㅋㅋㅋ 한거 ㅈ또 없음
LGC	LGC	대해 더 깊이 있게 설명해주면 이해하기 쉬울 것 같은데요.
		이런 만남은 정말 큰 의미가 있다고 생각합니다.
HWC	LGC	사칭 계정 만드는 사람들은 왜 그럴까요, 이해를 할 수가 없어서
		포항 소식 뉴스로 접하고 마음이 아팠어요.
LGC	HWC	총선에서 완전 괴멸로 만들어 줄께
		[redacted]가 남들은 소설써서 욕하고 매장 시키더니

Table 14: Visualization examples of detection model analysis using XAI.

This transformation enhances detection by maintaining key stylistic features without losing structural integrity, making it particularly effective for distinguishing HWC and LGC.

E.4 Visualization of Detection Model Analysis

Table 14 presents a visualization of the detection model's decision-making process using XAI. The highlighted text segments indicate key linguistic features that influenced classification, with red representing features indicative of LGC and green representing features typical of HWC.

The examples illustrate various prediction outcomes, including correct classifications and misclassifications. The model effectively identifies characteristic LLM expressions, such as formal structures and neutral phrasing, as well as human-specific informal markers like repeated characters (e.g., "ㅋㅋㅋㅋ"), conversational tones, and emotional expressions.

By integrating 300 special tokens derived from XAI analysis, our tokenizer enhances detection accuracy by capturing stylistic differences between LGC and HWC. This visualization highlights the importance of linguistic patterns in distinguishing human-authored and AI-generated comments.

F Experimental Setup

F.1 Data Collection for Detection

To train the detection model $D(x)$, we curated a large-scale dataset comprising both *LGC* and *HWC*. Given the potential presence of LGC in post-2023 news comments due to the widespread use of LLMs, we ensured dataset integrity by collecting HWC from periods before LGC became prevalent.

HWC Data Collection: For Korean HWC, we collected 1.3M comments from 135K news articles published in 2022 on *major Korean news platforms*. Data collection focused on high-profile news channels, selecting only posts with at least 15 comments. Each comment met the criteria of having a minimum of 15 characters or 4 words and a maximum of 280 characters or 50 words.

LGC Data Generation: LGC was generated using the *LGC generation framework* to ensure diversity in generation conditions. We created 1.8M LGC using 14 distinct LLMs.

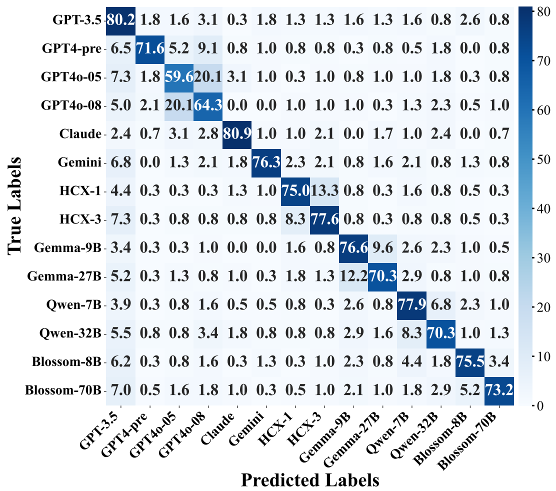
Validation and Test Set Construction: The validation and test sets contain 10K samples, maintaining a 1:1 ratio of HWC to LGC. The LGC subset in both datasets was stratified across LLM models, temperature settings, and sentiment to ensure balanced representation.

Evaluation Based on Comment Length: Most LLM-generated text detection models impose minimum length restrictions, such as GPTZero (250 characters) and DNT-GPT (180–300 words). However, real-world comments are significantly shorter, averaging 51 on Korean news platforms. This discrepancy highlights a fundamental limitation of existing detection methods, which are not optimized for short-form text. To address this gap, we evaluate detection performance across different comment lengths by dividing the test data into three categories: long (words > 12), medium (words 6–12), and short (words < 6). This analysis provides insights into the model's ability to detect LLM-generated comments in contexts where traditional approaches often fail.

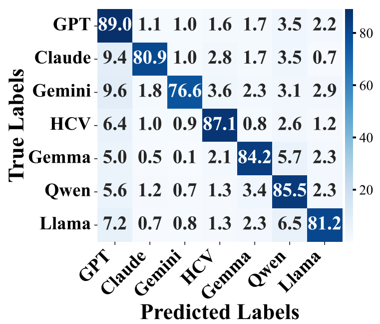
F.2 Data Collection for Attribution

LGC Attribution Dataset: For our LLM attribution experiments, we constructed a comprehensive dataset consisting of 1M LLM-generated comments created using 14 distinct language models across major AI providers. These models represent the current state-of-the-art in language generation and include: **GPT** (gpt-3.5-turbo-0125, gpt-4-0125-preview, gpt-4o-2024-05-13, gpt-4o-2024-08-06), **Claude** (claude-3-5-sonnet-20240620), **Gemini** (gemini-10-pro), **HCX** (HCX-DASH-001, HCX-003), **Gemma** (gemma-2-9b-it, gemma-2-27b-it), **Qwen** (Qwen2.5-7B-Instruct, Qwen2.5-32B-Instruct), **Llama** (llama-3-Korean-Bllossom-8B, llama-3-Korean-Bllossom-70B).

We structured this dataset as a 14-class classification task, with each class corresponding to a specific LLM model. To ensure robust evaluation, we created separate validation and test sets, each containing 5K samples. These sets were carefully curated to maintain complete separation from the training data while preserving balanced representation across three key dimensions: LLM models, temperature settings, and sentiment distributions. This balanced design ensures that our evaluation accurately reflects real-world attribution challenges and provides a reliable benchmark for assessing attribution performance.



(a) Korean LLM Attribution (Model-Level)



(b) Korean LLM Attribution (Family-Level)

Figure 10: Confusion matrices for Korean LLM attribution. (a) Attribution at the individual model level, where models from the same family (e.g., GPT-series, Llama-series) show higher misclassification. (b) Attribution at the LLM family level, which reduces confusion and improves classification performance.

Confusion Matrix Analysis and Model Grouping for Attribution: Initial analysis of the confusion matrices in Figure 10(a) revealed that while most models were well distinguished, models from the same family (e.g., GPT-series, Llama-series, Qwen-series) exhibited higher misclassification rates. This indicates that LLMs within the same family share stylistic and structural similarities that make distinguishing them more challenging. To address this, we grouped LLMs by provider (GPT, Claude, Gemini, HCX, Gemma, Qwen, Llama) and performed LLM family-level attribution instead of individual model-level attribution. The results in Figure 10(b) show that this grouped approach significantly improved performance by reducing confusion between closely related models. Additionally, we applied adjusted thresholding for each LLM family and measured the resulting F1-score improvements, demonstrating the effectiveness of hierarchical attribution over single-model classification. By incorporating both model-level and family-level attribution, we achieved more robust LLM attribution performance, mitigating confusion between structurally similar LLMs while maintaining fine-grained classification where possible.

F.3 Adversarial Text Manipulation Methods

To evaluate XDAC's robustness against humanization strategies that make LLM-generated comments appear more human-like, we applied two transformation methods and assessed their impact on detection accuracy, as illustrated in Figure 11.

Original Comment: "좋은 기사 감사합니다! 앞으로도 계속 응원할게요" ("Thank you for the great article! I will continue to support you")

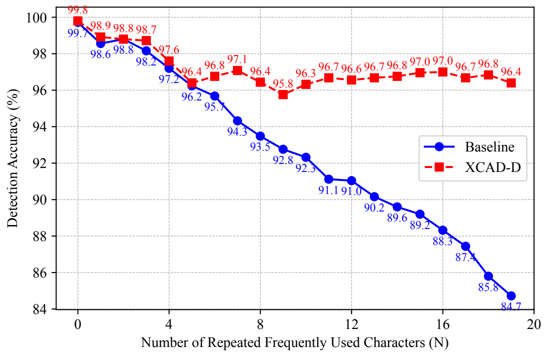
Adding Repeated Characters Frequently Used by Humans Humans often append frequently used characters, such as "ㅋ" (laughter) and "ㅎ" (soft chuckle) to emphasize tone or emotion. This experiment involved adding these characters at the beginning or end of the comment, with repetition levels ranging from 0 (original) to 20. The highlighted characters indicate the additional adversarial characters inserted into the original text as part of the attack strategy.

Example: "좋은 기사 감사합니다! 앞으로도 계속 응원할게요 ㅋㅋㅋㅋㅋ"

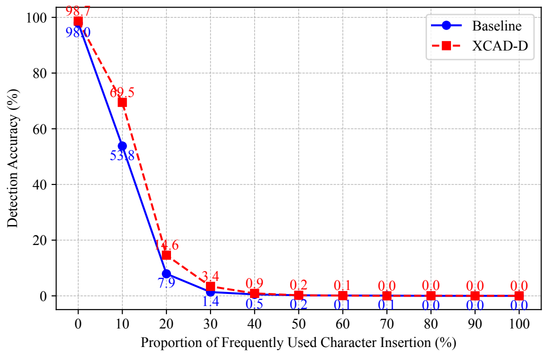
Inserting Frequently Used Human Characters Instead of appending characters at the end, this strategy inserts frequently used human-like characters ("ㅋ", "ㅎ", ".", "!", "?") between each character within the text. The insertion ratio varied from 0% (original) to 100%, where 50% means the characters were inserted in half of the possible positions.

Example: "좋은 기.사ㅋ 감사합?니다! 앞으로도 !계속. 응원할ㅋ게ㅋ요."

XDAC exhibited strong resistance to character-based transformations, including adding repeated human-used characters and inserting frequently



(a) Adding Repeated Characters That Are Frequently Used by Humans



(b) Inserting Frequently Used Human Characters

Figure 11: Impact of humanization strategies on LLM-generated comment detection.

used human characters, demonstrating its robustness against such modifications.