KOTOX: A Korean Toxic Dataset for Deobfuscation and Detoxification

Anonymous ACL submission

Abstract

Online communication increasingly amplifies toxic language, and recent research actively explores methods for detecting and rewriting such content. Existing studies primarily focus on non-obfuscated text, which limits robustness in the situation where users intentionally disguise toxic expressions. In particular, Korean allows toxic expressions to be easily disguised through its agglutinative characteristic. However, obfuscation in Korean remains largely unexplored, which motivates us to introduce a **KOTOX**: Korean toxic dataset for deobfuscation and detoxification. We categorize Korean obfuscation patterns into linguistically grounded classes and defines transformation rules derived from real-world examples. Using these rules, we provide paired neutral and toxic sentences alongside their obfuscated counterparts. Models trained on our dataset better handle obfuscated text without sacrificing performance on non-obfuscated text. This is the first dataset that simultaneously supports deobfuscation and detoxification for the Korean language. We expect it to facilitate better understanding and mitigation of obfuscated toxic content in LLM for low-resource languages.

1 Introduction

Throughout human history, toxic expressions have consistently appeared in communication, and detecting such expressions has long been recognized as an ethically significant challenge. With the advent of Language Models (LM), research has shifted from traditional rule-based methods to LM-driven approaches that leverage their language comprehension abilities to detect toxic text (Kim et al., 2024; Ahn et al., 2024; Kim et al., 2023; Lee et al., 2025; Hartvigsen et al., 2022a). Recently, researchers have increasingly focused on detoxification, which rewrites toxic text into non-toxic alternatives (Huimin et al., 2025; Ko et al., 2025; Tang et al., 2023).

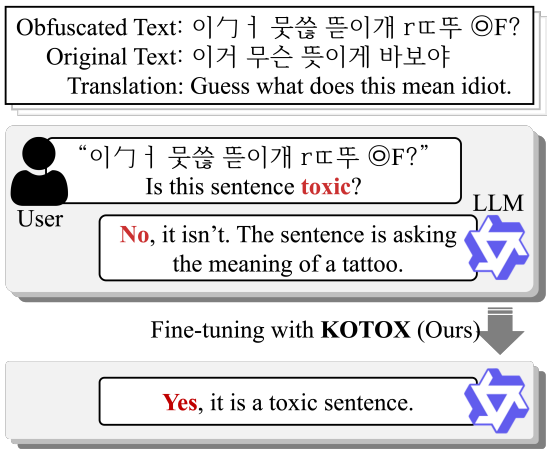

Figure 1: Comparison of obfuscated toxic text detection results before and after fine-tuning on KOTOX.

Meanwhile, users intentionally obfuscate toxic expressions to evade automatic moderation systems. Such obfuscation modifies surface forms while preserving the original intent, which complicates reliable detection. Several studies investigate this challenge by evaluating model robustness to textual perturbation in toxicity detection. Works such as Xiao et al. (2024a) and Röttger et al. (2021) show that minor typographical or orthographic alterations can severely degrade toxicity detection performance of models, revealing vulnerabilities of language models to obfuscated inputs. These findings indicate that obfuscation poses a substantial challenge for current toxicity detection models.

Most existing toxicity datasets and benchmarks focus on non-obfuscated text (ElSherief et al., 2021; Hartvigsen et al., 2022b). Moreover, existing obfuscation approaches rely on simple techniques such as homophone replacement or emoji insertion (Wei et al., 2024; Zhang, 2025; Xiao et al., 2024b). In addition, existing resources do not provide jointly aligned toxic content with its obfuscated variants, which makes unified experimentation difficult.

Dataset	Lang.	Toxic	Obfus.	Pair Type	Size	Obfus. Types
SBIC (Sap et al., 2020)	EN	O	X	–	44.0K	–
CADD (Song et al., 2021)	EN	O	X	–	24.5K	–
ToxiGen (Hartvigsen et al., 2022c)	EN	O	X	–	274.2K	–
KOLD (Jeong et al., 2022)	KO	O	X	–	40.4K	–
ParaDetox (Logacheva et al., 2022)	EN	O	X	$n \leftrightarrow t$	12.6K	–
K/DA (Jeon et al., 2025)	KO	O	X	$n \leftrightarrow t$	7.5K	–
HateCheck (Röttger et al., 2021)	EN	O	O	–	3.7K	PHON
ToxiCloakCN (Xiao et al., 2024a)	ZH	O	O	$t \leftrightarrow t^{(o)}$	1.5K	PHON / ICON
KOTOX (Ours)	KO	O	O	$n \leftrightarrow t$, $n \leftrightarrow n^{(o)}$, $t \leftrightarrow t^{(o)}$	6.9K	PHON / ICON / TRANS / SYN / PRAG

Table 1: Representative toxic datasets. *Obfus.* denotes datasets containing obfuscated toxic content. *Pair Type* indicates the pairing scheme, where n = neutral, t = toxic, and the $^{(o)}$ marks obfuscated forms. *Obfus Types* represent the applied obfuscation approaches: phonological, iconological, transliteration-based, syntactic, and pragmatic.

In particular, Korean is an agglutinative language with flexible spacing and rich morphological variation (Sohn, 1999; Taylor and Taylor, 2014) which allows surface forms to change without disrupting meaning. Its writing system further enables obfuscation through phonological variation and visual similarity that remain easily interpretable to native speakers. These linguistic characteristics lead to diverse and systematic, obfuscation patterns in real-world usage. Despite this, obfuscation in Korean toxic text remains relatively underexplored in existing research.

In response to these limitations, we introduce **KOTOX**, a Korean Toxic dataset designed for deobfuscation and detoxification. We organize Korean obfuscation into linguistically grounded classes, and defines transformation rules derived from real-world instances. By using these rules, we provide paired neutral and toxic sentences along with their obfuscated counterparts, which allows models to learn both text recovery and toxic rewriting.

We support three evaluation tasks: (i) Obfuscated Toxic Text Classification, (ii) Neutral Text Deobfuscation, and (iii) Obfuscated Toxic Text Sanitization. We evaluate these tasks using multiple toxicity classifiers and large language models under zero-shot, few-shot, and fine-tuning settings. The results show that training with KOTOX improves robustness to obfuscated toxic text while preserving performance on non-obfuscated inputs. To the best of our knowledge, KOTOX is the first high-quality paired dataset of obfuscated Korean toxic text. We expect KOTOX to facilitate a deeper analysis of obfuscated toxic content in Korean.

2 Related Works

2.1 Toxicity Classification

Early studies on toxic text classification primarily employed lexical or keyword-based approaches (Waseem et al., 2017; Ocampo et al., 2023). The development of deep learning accelerated the creation of various toxic datasets for model training. Representative datasets such as SBIC (Sap et al., 2020) and ToxiGen (Hartvigsen et al., 2022c) cover a wide spectrum of abusive, hateful, and biased texts collected from social media. For toxicity classification, previous research explored encoder-based fine-tuning approaches (Caselli et al., 2021; Liu et al., 2019; Wan et al., 2022), as well as contrastive learning methods (Kim et al., 2022; Ahn et al., 2024).

2.2 Detoxification

Unlike classification, detoxification requires rewriting toxic text into a neutral counterpart while preserving its semantic content. Motivated by the need, paired corpora such as ParaDetox (Logacheva et al., 2022) and K/DA (Jeon et al., 2025) provide parallel toxic-neutral sentences for model supervision. Meanwhile, these paired corpora are utilized to train models that rewrite toxic language into neutral forms, or to suppress toxic content generation during decoding (Ko et al., 2025).

2.3 Obfuscated Toxicity

In recent years, researchers recognized the need to evaluate model robustness against intrinsically complex or intentionally obfuscated toxic language. Within this line of work, several studies focused on obfuscation-based robustness. HateCheck (Röttger et al., 2021) employed leetspeak or orthographic

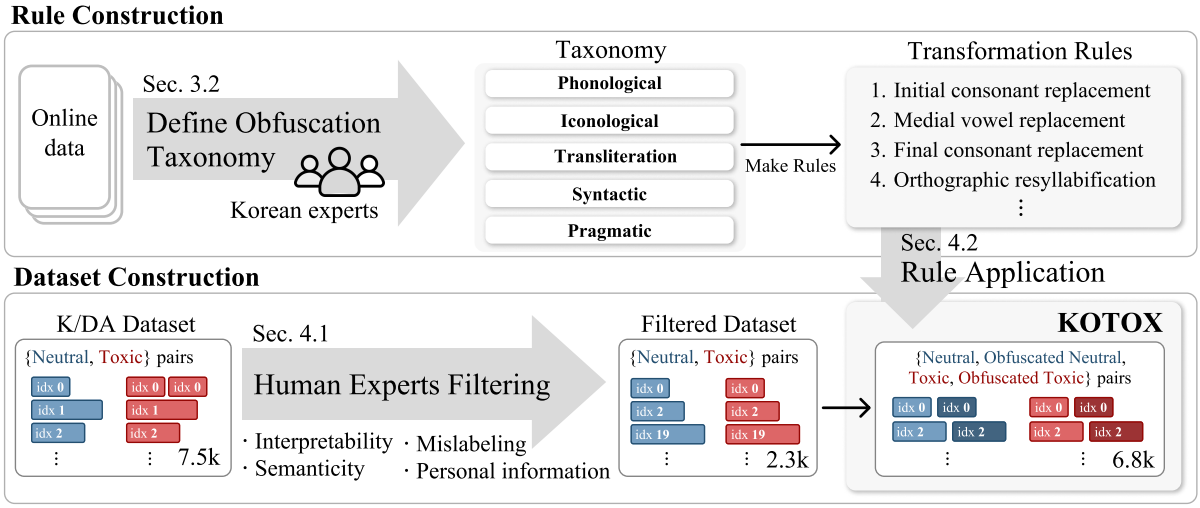

Figure 2: Overview of KOTOX construction pipeline. It encompasses the design of transformation rules, source corpus filtering of neutral-toxic pairs, and the generation of quadrupled obfuscated variants for each sample.

perturbations to challenge toxic detection models, while ToxiCloakCN (Xiao et al., 2024a) showed that homophone and emoji substitutions in Chinese substantially degrade model performance. Together, these findings indicate that even surface-level obfuscation can effectively undermine both toxicity detection and detoxification systems.

2.4 Limitations of Previous Works

Existing toxic datasets exhibit two key limitations. First, they address either non-obfuscated toxic texts for detoxification or obfuscated toxic texts for detection in isolation, leaving no dataset that jointly captures both toxicity and obfuscation. Second, they mainly target narrow surface changes (e.g., homophones, or emojis substitutions), yielding limited variety. These limitations highlight the need for a paired, obfuscation-aware dataset that includes both neutral and toxic texts along with their obfuscated counterparts. Such a resource enables integrated evaluation and training on toxicity and obfuscation within a unified framework.

3 Overview of KOTOX & Tasks

We introduce a **KOTOX**, a Korean neutral-toxic pair dataset that includes corresponding obfuscated counterparts. We extend obfuscation beyond simple spelling or visual modifications in prior work by leveraging linguistic properties of Korean and its writing system, Hangeul. Korean is an agglutinative language, and Hangeul is a compositional script in which a syllable block decomposes into three parts (e.g., ㅊ+ ㅐ+ㄱ → 책). This structure enables fine-grained phonological and iconological transformations. The resulting pairs constitute a challenging benchmark for robustness analysis.

3.1 Task Definitions

We define three tasks that jointly address toxicity and obfuscation, enabled by the KOTOX dataset. These tasks are more challenging than conventional settings and can be utilized for evaluating the robustness of LLMs.

Obfuscated Toxic Text Classification Given an obfuscated text, the goal of the task is to classify whether the given text is toxic or not. This mirrors standard toxicity classification but explicitly evaluates robustness under obfuscation.

Neutral Text Deobfuscation Given an obfuscated neutral text, the goal of the task is to generate its deobfuscated neutral text. This task is newly defined in our work and can be regarded as a form of constrained translation.

Obfuscated Toxic Text Sanitization Given an obfuscated toxic text, the goal of the task is to generate the deobfuscated neutral text that preserves semantics while removing toxicity. This task combines detoxification and deobfuscation in one step—the most challenging setting supported in KOTOX.

3.2 Class of Korean Obfuscation

Figure 2 illustrates our approach to construct a Korean obfuscation dataset. We classify obfuscation methods into five categories based on the linguistic taxonomy of Korean. Two Korean experts

Category	Rule
Phono-logical	1. Initial consonant replacement 2. Medial vowel replacement 3. Final consonant replacement 4. Orthographic resyllabification 5. Initial consonant insertion 6. Medial vowel insertion 7. Final consonant insertion 8. Liaison (forward, reverse)
Icono-logical	9. Hangeul look-alike 10. Cross-script substitution 11. Rotation-based variation
Trans-literation	12. Phonetic substitution (Latin) 13. Phonetic substitution (CYK) 14. Semantic substitution
Syntactic	15. Spacing perturbation 16. Syllable anagram
Pragmatic	17. Symbol/emoji insertion

Table 2: Transformation rules grouped by category with rule indices. Details of rules and examples are presented in Appendix B.

analyze 144 real-world obfuscated instances collected from user reviews on platforms (e.g., Agoda, Google Maps, and Booking.com), and they identify recurring obfuscation characteristics used by native speakers. We organize these characteristics into a structured taxonomy and define 17 transformation rules accordingly, as summarized in Table 2.

Phonological approach. We adopt a phonological approach that treats phonemes as the smallest units of sound. Korean exhibits unique phonological properties, where small textual changes yield diverse but phonetically similar sounds. This characteristic enables obfuscation by replacing words with phonetically close alternatives or by modifying text to match actual pronunciation. We define 8 rules for this phonological process, and Appendix B.1 describes these rules in detail.

Iconological approach. The iconological approach converts text by leveraging visual similarity. It substitutes characters with visually analogous symbols, numbers, or foreign scripts, such as Chinese characters. Hangeul, the Korean writing system, consists of syllabic blocks that can be decomposed into up to three components. These transformations preserve readability while introducing iconological variation. We establish three rules for this process, detailed in Appendix B.2.

Transliteration-based approach. This approach converts text into another language that shares the same pronunciation. Obfuscation occurs when Korean pronunciation is transcribed into English letters or replaced with Chinese characters that sound identical. Alternatively, obfuscation can be achieved by translating a Korean word into a foreign-language synonym and then phonetically transcribing it into Hangeul. We specify three rules for this approach, presented in Appendix B.3.

Syntactic approach. The syntactic approach operates at the word and sentence levels rather than at the character level. Korean differs from English due to its agglutinative morphology and grammar-dependent spacing, which allows deviations to obscure meaning. Korean word recognition relies on holistic syllabic blocks rather than sequential phonemes, which enables readers to infer meaning even when internal character order changes. We exploit these linguistic characteristics to establish two transformation rules. The details of the rules appear in Appendix B.4.

Pragmatic approach. This process perturbs text by inserting irrelevant elements, such as symbols or onomatopoeia. Prior work reports that adding such elements can evoke positive sentiment, thereby reducing the effectiveness of toxicity detection in large language models (Röttger et al., 2021). Appendix B.5 provides details of the rule-based pragmatic approach.

4 KOTOX Construction

Figure 2 illustrates the overall process of our data construction. Based on the previously defined rules, we construct neutral-toxic paired data containing corresponding obfuscations, enabling three tasks: obfuscated toxic text classification, neutral text deobfuscation, and obfuscated toxic text sanitization.

4.1 Source Dataset Preprocessing

We use the K/DA dataset (Jeon et al., 2025), consisting of Korean neutral-toxic sentence pairs, as the source corpus for constructing KOTOX dataset. We identify several quality issues within the original data, including imbalance, misaligned neutrality, semantic ill-formedness, and ethical concerns such as the exposure of personal information. For a reliable alignment, three Korean natives conducted a

Algorithm 1 Neutral-toxic pair obfuscation

Input: Neutral-toxic pair (x^n, x^t), rule set $\mathcal{R}$, rewrite rate $M = \{r : \tau_r\}_{r \in \mathcal{R}}$, apply number k

Output: Obfuscated pair (x^n, x^t), applied rules Π

```
Π ← ∅
for i = 1 to k do
    while R ≠ ∅ do
        r ← SAMPLE(R); τ ← M[r]
        y^n ← APPLYRULE(x^n, r, τ)
        y^t ← APPLYRULE(x^t, r, τ)
        if SANITYCHECK(y^n, y^t, Π, r) then
            x^n ← y^n,  x^t ← y^t
            Π ← Π ∪ {r}
            break;
        end if
        R − {r}
    end while
end for
return (x^n, x^t), Π
```

manual filtering process based on a 10-item rubric covering label fidelity, linguistic validity, and data distribution integrity. The experts independently reviewed 7,555 pairs, achieving a Gwet's AC1 score of 0.7408 ($p < 0.001$), indicating inter-annotator agreement. This rigorous refinement yielded 2,294 high-quality pairs, ensuring a reliable and balanced foundation for the KOTOX. The details appear in Appendix C.1.

4.2 Construct Obfuscation of Text

Using the filtered neutral-toxic pairs, we construct KOTOX by applying the implemented transformation rules to each pair. For every source pair, three augmented pairs are generated by repeating the rule-application process $k \in \{2, 3, 4\}$ times.

As shown in Alg. 1, given a single pair, the algorithm samples a rule r from the rule set $\mathcal{R}$ and applies it to both the neutral and toxic sides. If the applied result violates any sanity check (SANITYCHECK), a new rule is resampled and reapplied until successful modification is achieved. This mechanism ensures that each pass introduces a meaningful transformation and avoids trivial or destructive overlaps among rules.

4.3 Dataset statistics

The dataset follows an 8:1:1 split for training, validation, and testing, resulting in 5,505 training instances, 687 validation instances, and 690 test instances. The details of dataset statistics are presented in Appendix C.4.

5 Experimental Settings

5.1 Classification

To investigate the detection capability of different models, we conduct a toxic text classification task. We compare model performance on non-obfuscated dataset (Toxic) and obfuscated dataset (Ours) using F1-score to examine their understanding of obfuscated toxic content. For LMs, we perform supervised-fine-tuning (SFT) independently on each dataset and conduct cross-validation across them. For LLMs, we apply few-shot prompting using examples from the corresponding datasets.

Classification models. We use three LMs fine-tuned on toxic datasets for the classification task, along with one open-source and one closed-source LLM. HateBERT[1] was fine-tuned on Reddit posts, offensiveRoBERTa[2] (RoBERTa) was fine-tuned on Kaggle toxic comment challenge dataset, and toxicity-xlmr-v2[3] (XLM-R) was fine-tuned on multilingual corpora covering 15 languages from various language families. Qwen2.5[4] is a strong multilingual instruction-tuned LLM and GPT-4.1 is a closed-source LLM representing the proprietary models.

5.2 Deobfuscation and Sanitization

For the Deobfuscation and Sanitization tasks, we perform experiments in two settings: LLM prompting and fine-tuning. The experiments consist of zero-shot prompting, five-shot prompting, and SFT. We train the SFT models using LoRA and repeat each experiment three times for consistency. Detailed configurations are provided in Appendix D.

LLMs. We employ four LLMs selected to ensure linguistic diversity. The open-source set comprises Qwen2.5, along with two Korean-focused LLMs, EXAONE 3.5[5] and Bllossom[6]. These three models have comparable parameter sizes and are all instruction-tuned. We also use GPT-4.1.

Toxicity & similarity metrics. We report evaluation results using two common metrics for both

[1] GroNLP/hateBERT
[2] unitary/multilingual-toxic-xlm-roberta
[3] textdetox/xlmr-large-toxicity-classifier-v2
[4] Qwen/Qwen2.5-7B-Instruct
[5] LGAI-EXAONE/EXAONE-3.5-7.8B-Instruct
[6] MLP-KTLim/llama-3-Korean-Bllossom-8B

Model	Eval	Base	Toxic	Ours	Comb.
HateBERT (LM)	Toxic	36.56	76.69	77.19	**78.44**
	Ours	36.28	65.88	**71.65**	71.32
	Δ	0.28	10.81	**5.54**	7.12
RoBERTa (LM)	Toxic	33.29	91.86	92.02	**92.68**
	Ours	33.61	69.98	84.97	**86.94**
	Δ	-0.32	21.88	7.04	**5.74**
XLM-R (LM)	Toxic	79.28	95.06	**96.30**	96.16
	Ours	56.80	53.66	**89.57**	88.13
	Δ	22.48	41.40	**6.73**	8.03
Qwen2.5 (LLM)	Toxic	**83.66**	79.00	82.32	83.13
	Ours	69.01	69.85	70.03	**70.32**
	Δ	14.65	**9.15**	12.29	12.13
GPT-4.1 (LLM)	Toxic	89.47	**92.05**	90.21	91.80
	Ours	78.34	**80.93**	80.56	80.65
	Δ	11.13	11.12	**9.65**	11.15

Table 3: Binary toxicity classification under obfuscation (F1-score (%)). For LMs, the Base setting indicates no fine-tuning, while for LLMs, it indicates zero-shot inference. Comb. denotes the combination of Toxic and Ours, along with the robustness gap Δ =Toxic−Ours. The best performance and the smallest gap are highlighted in bold.

deobfuscation and sanitization, and one additional metric for sanitization. To measure similarity with the reference text, we use BertScore (Zhang et al., 2020) and chrF (Popović, 2015). To evaluate the toxicity of sanitized outputs, we employ Google Jigsaw's Perspective API[7], which is widely adopted in detoxification tasks.

6 Experimental Results

6.1 Obfuscated Toxic Text Classification

Table 3 presents the result of the toxic classification task. Since none of the three LMs were pretrained on Korean data, the performance without tuning (Base) is considerably low. However, XLM-R achieves relatively higher performance due to multilingual pretraining. Models fine-tuned only on the non-obfuscated toxic dataset (Toxic) shows substantially lower performance on the obfuscated evaluation set (Ours) than on the Toxic set. This result indicates that understanding toxic expressions alone is insufficient for detecting obfuscated toxic text.

[7] https://perspectiveapi.com/

Models trained exclusively on Ours set achieve higher performance on the Toxic evaluation set than models trained only on the Toxic set. Moreover, the performance gap between the Ours and Toxic evaluations decrease by up to 34.67%p. Finally, when the models were trained on both the Toxic and Ours set (Comb.), the results were comparable to those obtained when using only our dataset. This confirms that our dataset enhances the detection of obfuscated toxic text without degrading performance on the original toxic data.

For LLMs, both Qwen2.5 and GPT-4.1 show stronger performance on the Toxic dataset in the zero-shot (Base) setting than the LM baselines, while their performance noticeably degrades on the obfuscated dataset. The five-shot results further indicate that few-shot prompting does not consistently improve performance, and the degree of improvement differs across models and datasets. GPT-4.1 benefits from few-shot prompting more reliably than Qwen2.5, which suggests that Korean obfuscation affects LLMs in different ways. These findings show that the proposed benchmark captures differences in obfuscation robustness across models and reveals variation in their ability to interpret obfuscated Korean text.

6.2 Neutral Text Deobfuscation

Table 4 shows the experimental results for deobfuscating obfuscated neutral texts. We conduct experiments under three configurations: zero-shot, five-shot, and supervised fine-tuning (SFT). In the zero-shot setting, all models exhibit lower deobfuscation performance, even though they are pretrained on Korean dataset. In the five-shot setting yields small improvements in BERTScore across models. All open-source models achieve their best performance under the SFT setting.

The chrF score, an n-gram-based metric, increases substantially compared to the zero-shot results under SFT. The BERTScore, which measures semantic similarity based on embedding, shows improvements of up to 11%p SFT. The closed-source model GPT achieves the highest overall performance. These results suggest that existing LLMs, which are typically trained on clean and noise-free text, have limited understanding of obfuscated Korean text. By contrast, models fine-tuned on our dataset acquire a better understanding of obfuscation patterns, demonstrating improved robustness and comprehension of obfuscated Korean toxic texts.

Setting	Qwen2.5		EXAONE3.5		Bllossom		GPT-4.1	
	BertScore	chrF	BertScore	chrF	BertScore	chrF	BertScore	chrF
Zero-Shot	65.96	15.31	60.60	7.64	65.09	14.08	83.17	41.77
Five-Shot	68.93	19.40	67.00	14.39	70.02	21.14	87.22	52.62
SFT	77.90	36.32	78.12	34.39	78.05	39.97	-	-

Table 4: Neutral text deobfuscation experiment result. We use three open-source LLMs and one closed LLM. The table shows the performance on the settings of zero-shot, five-shot, and fine-tuning.

Shots	Qwen2.5			EXAONE3.5			Bllossom			GPT-4.1		
	Bert.	chrF	Pers.	Bert.	chrF	Pers.	Bert.	chrF	Pers.	Bert.	chrF	Pers.
Zero	62.48	7.30	9.89	58.34	3.47	7.87	58.69	3.91	12.58	73.39	16.48	6.91
Five	65.70	10.11	11.51	63.67	6.87	8.49	66.11	11.03	13.29	76.78	23.07	7.35
SFT	71.03	15.06	4.35	71.17	13.53	6.38	70.92	16.31	4.31	-	-	-

Table 5: Toxic text sanitization experiment result. We use three open source LLMs and one closed LLM. The table shows the performance on the settings of zero-shot, five-shot, and finetuning. We additionally report the perspective API toxicity score. Lower values indicate lower toxicity in the Perspective API.

6.3 Obfuscated Toxic Text Sanitization

Figure 5 presents the results of transforming obfuscated toxic texts into deobfuscated neutral texts. The Sanitization task shows very low performance in the zero-shot setting, similar to the deobfuscation experiments.

In the five-shot setting shows slight improvements in BERTScore and chrF. However, the Perspective API scores increase in the five-shot setting, where higher values indicate higher toxicity. These results indicate that models often succeed in deobfuscation but fail to mitigate toxic content in the five-shot setting. Manual inspection of the generated outputs confirms that models recover surface forms while retaining toxic meaning in many cases. These observations suggest that five-shot prompting does not provide sufficient understanding of obfuscation for successful sanitization.

The SFT setting achieves the best performance, consistent with the deobfuscation results. Models fine-tuned on KOTOX show improved ability to interpret obfuscated sentences and generate non-toxic outputs. These results indicate that current LLMs still have limited understanding of obfuscated Korean text, making them highly vulnerable to obfuscated toxic content. Therefore, our dataset is essential for building models that are robust to toxicity and resilient against obfuscated language.

7 Dataset Analysis

7.1 Rule analysis

Figure 3 presents the classification error ratio of HateBERT fine-tuned on the KOTOX for each applied rule. The error ratio represents the proportion of incorrect predictions for each rule. Figure 11 illustrates the correlation among the rules. The rules exhibit very little correlation with one another, which allows each rule to be interpreted independently. Rule 15 corresponds to the spacing perturbation rule and shows the highest error ratio. Although spacing changes do not significantly affect human understanding of the original meaning, they severely impact LMs because the models process text at the token level. When token boundaries are disrupted, model performance becomes highly vulnerable. Rule 17, which is symbol/emoji insertion, also causes a high error ratio. These symbols are unrelated to the textual context and hinder the model's ability to detect toxicity. They can also induce misleadingly positive sentiment, thereby threatening the model's robustness. In contrast, within the phonological approach, rules such as 8, which are based on clearly defined pronunciation patterns, tend to yield lower error ratios. This suggests that LMs can more easily capture systematic phonological transformations than irregular or noise-like modifications.

Figure 3: Error ratio for each rule. HateBERT is trained and evaluated on the KOTOX datasets. The error ratio indicates the proportion of misclassified samples among the data associated with each rule.

	S1	S2	S3	Avg.	Qwen
Bert.	95.73	96.04	95.16	95.64	77.90
chrF	82.91	82.89	80.61	82.13	36.32

Table 6: Human deobfuscation evaluation results. S1, S2, and S3 denote the three Korean native speaker, and Qwen denotes Qwen2.5 fine-tuned on KOTOX.

7.2 Semantic Preservation

We conduct a human deobfuscation evaluation on 500 sampled KOTOX test set to verify whether sentence meaning remains preserved after applying transformation rules. Table 6 presents the results of the human evaluation and Qwen2.5 result fine-tuned on our dataset. Three native Korean speakers perform the deobfuscation task. Human evaluation achieves BERTScore values that are 17.75%p higher and chrF scores that are 45.81%p higher than those of the fine-tuned Qwen2.5 model, and it shows consistently strong performance in the 90% range. These results indicate that sentence meaning remains intact even under the application of many transformation rules. The high level of human performance indicates that the proposed rules are practically applicable. The comparison with current LLM performance show that existing LLMs still exhibit limited understanding of obfuscated Korean text.

7.3 Evaluation on Wild Data

We examine how models trained on KOTOX perform on wild data to evaluate their real-world generalization. We collect 144 obfuscated review instances from online platforms such as Agoda, Google Maps to construct the wild dataset. We conduct evaluation under zero-shot, five-shot, and supervised fine-tuning settings, where the five-shot settings use examples from KOTOX, and the supervised fine-tuning setting also fine-tunes Qwen2.5 on KOTOX. Table 7 presents the evaluation results.

Setting	KOTOX		Wild	
	Bert.	chrF	Bert.	chrF
Zero-Shot	65.96	15.31	63.03	11.36
Five-Shot	68.93	19.40	65.48	14.13
SFT	77.90	36.32	72.30	21.99

Table 7: Wild dataset evaluation with Qwen2.5. In the five-shot setting, we use examples from KOTOX, and in the supervised fine-tuning setting, we use Qwen2.5 fine-tuned on KOTOX.

The results show slightly lower performance on the wild dataset than on KOTOX, while overall performance patterns remain similar. This observation suggests that the wild dataset presents marginally higher difficulty than our dataset. At the same time, the consistent performance trends indicate that applying multiple transformation rules does not introduce excessive or unrealistic difficulty to the sentences. In the supervised fine-tuning setting, Qwen2.5 fine-tuned on our dataset outperforms the non-fine-tuned settings on the wild dataset, which indicates that training on our dataset helps the model better understand real-world obfuscated examples. These findings demonstrate that KOTOX captures real-world characteristics of Korean online communities.

8 Conclusion

In this paper, we propose **KOTOX**, a neutral-toxic paired dataset that includes obfuscated counterparts. We categorize obfuscation approaches into five classes based on Korean linguistic properties and define the corresponding transformation rules. By applying these rules, we construct a neutral-toxic paired dataset in which each instance includes its corresponding obfuscated counterpart. Using our dataset, we conduct classification, deobfuscation, and sanitization tasks, demonstrating that the dataset effectively facilitates these tasks. As far as we are aware, this is the first obfuscation and detoxification dataset in Korean, and we expect it will contribute to further research on improving the understanding of Korean obfuscation.

Limitations

Our study focuses exclusively on the Korean language and Hangeul. This design choice can be considered as both a limitation and a strength. KOTOX and its transformation rules may not directly generalize to other linguistic or cultural contexts. However, Korean presents unique phonological and orthographic characteristics that make obfuscation phenomena particularly rich and distinctive. Our dataset and analysis are therefore deliberately tailored to explore these language-specific traits in depth, providing insights that would be lost in a broad multilingual setting. In future work, we plan to extend the obfuscation taxonomy and data construction framework to other languages.

Ethical Considerations

Our work involves the collection and analysis of toxic and offensive language, which inherently raises ethical concerns. All toxic samples used in KOTOX originate from publicly available sources, and sensitive or personally identifiable information was carefully removed during data filtering by following the rubrics in Table 17 in Appendix. 4.1. While our dataset includes harmful expressions for research purposes, it is intended solely for academic use in developing safer and more robust language technologies. We strongly discourage any misuse of KOTOX or its contents for generating, amplifying, or spreading offensive material.

References

Hyeseon Ahn, Youngwook Kim, Jungin Kim, and Yo-Sub Han. 2024. Sharedcon: Implicit hate speech detection using shared semantics. In *Findings of the Association for Computational Linguistics, ACL*, pages 10444–10455.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2786–2794, Marseille, France. European Language Resources Association.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. *arXiv preprint arXiv:2109.05322*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022a. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 3309–3326.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022b. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022c. Toxigen: A large-scale machine-generated dataset for implicit and adversarial hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2367–2388, Dublin, Ireland. Association for Computational Linguistics.

LU Huimin, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2025. Unidetox: Universal detoxification of large language models via dataset distillation. In *The Thirteenth International Conference on Learning Representations*.

Minkyeong Jeon, Hyemin Jeong, Yerang Kim, Jiyoung Kim, Jae Hyeon Cho, and Byung-Jun Lee. 2025. K/DA: Automated data generation pipeline for detoxifying implicitly offensive language in Korean. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21404–21432. Association for Computational Linguistics.

Younghoon Jeong, Juhyun Oh, Jongwon Lee, Jaimeen Ahn, Jihyung Moon, Sungjoon Park, and Alice Oh. 2022. KOLD: Korean offensive language dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10818–10833.

Jaehoon Kim, Seungwan Jin, Sohyun Park, Someen Park, and Kyungsik Han. 2024. Label-aware hard negative sampling strategies with momentum contrastive learning for implicit hate speech detection. In *Findings of the Association for Computational Linguistics, ACL*, pages 16177–16188.

Youngwook Kim, Shinwoo Park, and Yo-Sub Han. 2022. Generalizable implicit hate speech detection using contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING*, pages 6667–6679.

Youngwook Kim, Shinwoo Park, Youngsoo Namgoong, and Yo-Sub Han. 2023. Conprompt: Pre-training a language model with machine-generated data for implicit hate speech detection. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 10964–10980.

Ching-Yun Ko, Pin-Yu Chen, Payel Das, Youssef Mroueh, Soham Dan, Georgios Kollias, Subhajit Chaudhury, Tejaswini Pedapati, and Luca Daniel. 2025. Large language models can become strong self-detoxifiers. In *Proceedings of the 2025 International Conference on Learning Representations*.

Yejin Lee, Joonghyuk Hahn, Hyeseon Ahn, and Yo-Sub Han. 2025. AmpleHate: Amplifying the attention for versatile implicit hate detection. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28862–28874.

Ping Liu, Varada Kolhatkar, and Joel Tetreault. 2019. Offenseval: Identifying and categorizing offensive language in social media. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 86–94, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

Nicolás Benjamín Ocampo, Ekaterina Sviridova, Elena Cabrio, and Serena Villata. 2023. An in-depth analysis of implicit and subtle hate speech messages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, pages 1989–2005.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58. Association for Computational Linguistics.

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, Online.

Ho Sohn. 1999. Min. 1999. the korean language. *Cambridge: Cambridge UP*.

Hoyun Song, Soo Hyun Ryu, Huije Lee, and Jong Park. 2021. A large-scale comprehensive abusiveness detection dataset with multifaceted labels from reddit. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 552–561, Online. Association for Computational Linguistics.

Zecheng Tang, Keyan Zhou, Juntao Li, Yuyang Ding, Pinzheng Wang, Bowen Yan, Rejie Hua, and Min Zhang. 2023. Cmd: a framework for context-aware model self-detoxification. *arXiv preprint arXiv:2308.08295*.

Insup Taylor and M Martin Taylor. 2014. Writing and literacy in chinese, korean and japanese.

Zhen Wan, Yuan Ding, Shuai Jiang, Xiaoyu Huang, and Qianqian Xie. 2022. Toxicity detection across languages with xlm-r and fine-tuning strategies. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH)*, pages 1–10, Seattle, Washington. Association for Computational Linguistics.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *ALW@ACL*, pages 78–84. Association for Computational Linguistics.

Zhipeng Wei, Yuqi Liu, and N Benjamin Erichson. 2024. Emoji attack: Enhancing jailbreak attacks against judge llm detection. *arXiv preprint arXiv:2411.01077*.

Yunze Xiao, Yujia Hu, Kenny Tsu Wei Choo, and Roy Ka-wei Lee. 2024a. Evaluating robustness of offensive language detection in chinese: The toxicloakcn dataset. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yunze Xiao, Yujia Hu, Kenny Tsu Wei Choo, and Roy Ka-wei Lee. 2024b. Toxicloakcn: Evaluating robustness of offensive language detection in chinese with cloaking perturbations. *arXiv preprint arXiv:2406.12223*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations (ICLR)*.

Yangshijie Zhang. 2025. Emoti-attack: Zero-perturbation adversarial attacks on nlp systems via emoji sequences. *arXiv preprint arXiv:2502.17392*.

Class	Mapped Feature (Appx)	Type
Phonological	Combinatorial Syllabary (§A.2.1)	Korean
Iconological	Visual Decomposability (§A.3.1)	Hangeul
Transliteration-based	Multiscript Familiarity (§A.2.2)	Korean
Syntactic	Syllable-Oriented Segmentation (§A.3.2)	Hangeul
Pragmatic	—	Language-agnostic

Table 8: Obfuscation classes and their enabling properties. Features are detailed in Appendix (§A.2, §A.3).

A Preliminary

A.1 Korean Language & Hangeul

Korean is an agglutinative and morphologically rich language in which grammatical relations are expressed through affixes and particles. Its writing system, Hangeul, is a compositional and featural phonemic script: each syllable block is formed by combining an initial consonant, a medial vowel, and an optional final consonant (e.g., ㅊ+ ㅐ+ㄱ → 책). This block-based structure allows fine-grained phonological and visual variations, making Korean particularly suitable for studying diverse obfuscation phenomena.

As shown in Table 8, the proposed obfuscation classes exploit inherent linguistic and orthographic properties of Korean and Hangeul. The compositional structure of syllables, visual regularity of graphemes, and multilingual familiarity shared by Korean users collectively enable diverse and controllable transformation strategies. These characteristics make Korean particularly suitable for studying systematic and fine-grained text obfuscation.

A.2 Korean Language-Specific Properties

A.2.1 Combinatorial syllabic phonology.

Korean phonology is organized around syllabic units by the combination of initial consonant, medial vowel, and final consonant. This block-based composition induces dense neighborhoods of near-homophones at the syllable level, further enriched by the lenis–aspirated–tense triplets (e.g., ㄱ/ㅋ/ㄲ, ㄷ/ㅌ/ㄸ) and pervasive liaison/coarticulation phenomena. As a result, preserving the global "sound impression" while altering one or more sub-syllabic elements is structurally easy and perceptually tolerable for human readers. These properties systematically increase the search space for sound-preserving edits (replacement, addition) without severely degrading legibility, which directly enables phonological obfuscation.

A.2.2 Latent multiscript competence.

Due to historical and educational exposure, Korean users routinely navigate multiple scripts (Hangeul, basic chinese character, and Latin alphabet), and are familiar with bidirectional phonetic transcription conventions. This latent multiscript competence supports intuitive cross-script rendering of Korean words and names, and facilitates obfuscation by swapping to visually or phonetically similar forms in other scripts (or by re-Hangeulization after translation). The community-level familiarity with such code-mixed writing (e.g., signage, names, media) lowers the cognitive cost of interpreting transliterations, thereby making transliteration-based obfuscation particularly viable.

A.3 Hangeul Orthographic Properties

A.3.1 Decomposability and visual iconicity.

Hangeul graphemes are explicitly decomposable into consonants and vowels within a square syllabic layout. The clear sub-graphemic structure, together with geometric regularities of the block, affords visually motivated substitutions at both the character and consonant levels and rotation-based variants. Human readers retain robust recognition under such geometric perturbations due to the script's iconic regularity and redundancy, which, in turn, makes iconological obfuscation effective.

A.3.2 Syllable-oriented segmentation

Hangeul is written in syllabic blocks, and Korean readers parse strings with strong syllable-level awareness. Combined with historically variable spacing practices and the grammatical role of postpositional particles, this yields high tolerance to segmentation perturbations and syllable-level rearrangements: many strings remain human-recoverable despite spacing noise or local anagrams. This property directly supports syntactic obfuscation that disrupts surface structure while preserving overall interpretability.

Category	Granularity	Examples
Replacement	Initial consonant	한국인들만 알아볼 수 → 한꾹인뜰만 알아뽈 쑤
	Medial vowel	태국 → 타이국, 강해짐 ↔ 강하이짐
	Final consonant	낡았습니다 → 낡앆슾니다 , 돈 ↔ 돉
	Resyllabification	할 짓이가 ↔ 할찌시가
Insertion	Initial consonant	많이 → 많휘, 안에 → 안네
	Medial vowel	거품 점수줘서→ 궈퓸 졈슈줘숴
	Final consonant	호스트 → 홋스트, 바깥 → 박깥
Liaison	Forward liaison	들어봐 → 드러봐, 할아버지 → 하라버지
	Reverse liaison	바보 → 밥오, 버블 → 벱을

Table 9: Examples of the **Phonological Approach**. Each rule edits sub-syllabic components of Hangeul while maintaining intelligibility through phonological alternations.

B Classes of Obfuscation

B.1 Phonological Approach

The phonological approach exploits the similarity in pronunciation between sounds, modifying the phonemic components of a syllable while preserving overall phonetic perception. Three types of edits are applied—replacement, addition, and liaison—each operating on the sub-syllabic structure of Hangeul. Deletions are not employed, as they tend to remove excessive information and distort readability. Because Korean exhibits systematic phonological alternations (liaison), these operations are especially effective for generating natural yet obfuscated variants. As noted in Appendix A.2.1, each syllable in Hangeul can be decomposed into multiple components, which facilitates diverse and fine-grained variations.

Replacement. We replace sub-syllabic units that share close phonetic features: (i) Initial consonant, (ii) Medial vowel, and (iii) Final consonant. Each is substituted with a phonetically similar unit so that the pronunciation remains recognizable. Additionally, (iv) orthographic resyllabification is applied, where syllables are recomposed according to common phonological rules to reflect natural sound shifts. Korean provides rich substitution options owing to its lenis–aspirated–tense triplets (e.g., ㄱ/ㅋ/ㄲ) and various semi-vowels and diphthongs, which enable fine-grained and diverse replacements. As shown in Table 10, representative phonological substitution dictionaries such as lenis–tense and lenis–aspirated mappings form the basis of these replacement rules.

Lenis→Tense	**Lenis→Aspirated**	**Vowel→Diph.**
ㄱ → ㄲ	ㄱ → ㅋ	ㅏ → ㅑ
ㄷ → ㄸ	ㄷ → ㅌ	ㅓ → ㅕ
ㅂ → ㅃ	ㅂ → ㅍ	ㅗ → ㅛ
ㅅ → ㅆ	ㅈ → ㅊ	ㅜ → ㅠ
ㅈ → ㅉ	ㅊ → ㅋ	ㅡ → ㅢ

Table 10: Representative phonological substitution dictionaries used in the **Phonological Approach**. Each column denotes a systematic replacement pattern among consonants or vowels. Diph. refers to the 'Diphthong'.

Insertion. Additions insert new phonemes while retaining the original pronunciation pattern. (i) Initial consonant insertion: the silent consonant 'ㅇ' allows prefixing repeated or weak consonant sounds without changing syllable integrity. (ii) Medial vowel insertion: Korean vowels include semi-vowels (e.g., ㅏ→ ㅑ, ㅜ→ㅟ) that can be naturally inserted to create similar but extended sounds. (iii) Final consonant insertion: since the final consonant position in Hangeul is optional, a new consonant can be appended—often drawn from the onset of the following syllable—to mimic natural articulation.

Liaison. Liaison refers to the phonological process where the final consonant of a syllable is carried over to the initial position of the next. We simulate this by two variations: (i) forward liaison and (ii) reverse liaison, which performs the inverse mapping to obscure standard pronunciation patterns. These operations reflect natural pronunciation flow while introducing subtle orthographic perturbations that remain intelligible to human readers.

Category	Granularity	Examples
Look-alike	Hangeul	귀엽다 → 커엽다, 멍멍이 → 댕댕이
	CJK	쭈꾸미 ↔ 卒꾸미, 국밥 ↔ 弓밥
	Latin Scripts	야구 ↔ OF구, 태평 ↔ EH평
	Multiscripts or emoji	참치 → え占치, 바꾸자 → ㊐꾸자
Rotation	90° rotation	비버 → 뜨또, 똥 → 버0
	180° rotation	눈물 → 룸곡, 아이폰 → 굺I어 ㅇ

Table 11: Examples of the **Iconological Approach**. Look-alike transformations operate at both the character and jamo levels, substituting visually similar glyphs across scripts (Hangeul, CJK, Latin, symbols, or emoji). Rotation-based rules alter glyph orientation (90° or 180°) to generate visually perturbed yet readable text.

B.2 Iconological Approach

The iconological approach leverages the visual decomposability of Hangeul consonants and the independence of their graphical forms. As discussed in Sec. A.3.1, the clear sub-graphemic structure of Hangeul, together with the geometric regularity of its syllabic blocks, enables visually motivated substitutions at both the character and consonant levels, as well as rotation-based variants. As illustrated in Table 11, Hangeul allows a variety of iconographic transformations owing to its syllabic block structure and clear geometric regularity. These transformations are designed to modify the visual appearance of text while maintaining overall recognizability to human readers.

Look-alike substitution. This method substitutes Hangeul characters with visually similar glyphs. These substitutes can be other Hangeul characters or visually analogous symbols drawn from CJK (Chinese, Japanese, Korean) characters, Latin scripts, or even emojis.

Specifically, these substitutions occur at two different levels of granularity: (i) at the *character level*, entire syllable blocks are replaced with visually similar symbols. This is particularly frequent among Hangeul variants, emojis, and CJK characters. Due to their visual complexity, CJK characters are often effective at mimicking the overall structure of a complete Hangeul syllable. (ii) at the *sub-syllabic level*, individual graphemes (consonants and vowels) are replaced with shape-correlated symbols. For instance, the Hangeul letter ‘ㅇ’ can be replaced by the Latin ‘O’, or ‘ㅑ’ by ‘F’. Because Hangeul is a featural script where consonants and vowels are combined into blocks, this sub-syllabic structure allows for highly flexible and diverse look-alike substitutions.

Han.→Han.	Han.→CJK	Sub-syllabic
귀 → 커	틋 → 長	ㄱ ↔ プ
멍 → 댕	국 → 弓	ㄴ ↔ レ
비 → 네	흡 → 音	ㄷ ↔ て
면 → 띤	쭈 → 卒	ㄹ ↔ 己
명 → 띵	쇼 → 企	ㅁ ↔ 口
유 → 윾	슥 → 今	ㅂ ↔ せ
우 → 윽	리 → 引	ㅅ ↔ 人
점 → 겸	튼 → 長	ㅇ ↔ ○
과 → 파	숲 → 金	ㅈ ↔ 久
괄 → 팔	흠 → 高	ㅊ ↔ 大
관 → 판	매 → 叫	ㅋ ↔ ヲ
대 → 머	조 → 丕	ㅌ ↔ 巨
왕 → 앟	쇼 → 企	ㅍ ↔ 立
공 → 끙	몸 → 呂	ㅎ ↔ 云

Table 12: Representative iconological substitution dictionaries used in the **Iconological Approach**. Each column shows systematic visual mappings between (i) Hangeul–Hangeul replacements, (ii) Hangeul–CJK substitutions, and (iii) sub-syllabic correspondences. Han. denotes Hangeul.

Rotation. Rotation-based obfuscation manipulates the glyph orientation of Hangeul characters. By rotating syllable blocks or subcomponents by 90° or 180°, we produce text that visually resembles the original while disrupting standard orthographic patterns. Such geometric perturbations preserve readability to humans but often confuse automatic recognition models. For example, a 90° rotation of the Hangeul ‘비’ results in ‘뜨’, creating a visually similar but semantically different character.

Category	Granularity	Examples
Phonetic Transliteration	CJK substitution	수상해 → 水상해, 남한테 → 男한테
	Latin substitution	망했다고 → mang했다고, 게시판 → g 게시판
Semantic Transliteration	English meaning	가지 말고 같이 먹자 → 돈트 고 같이 먹자
	Japanese meaning	자리 좀 부탁해 → 자리 좀 구다사이

Table 13: Examples of the **Transliteration-based Approach**. Phonetic transliteration replaces parts of Hangeul words with phonetically similar units in CJK or Latin scripts, while semantic transliteration substitutes words with phonetic renderings of their foreign-language meanings (e.g., English or Japanese).

B.3 Transliteration-based Approach

As discussed in Sec. A.2.2, Korean users are inherently familiar with multiple writing systems, including Hangeul, basic Chinese characters (Hanja), and the Latin alphabet, due to historical and educational exposure. This multilingual competence enables intuitive transliteration-based obfuscation, where parts of text are replaced with characters or sounds drawn from other scripts that share phonetic or semantic associations. Broadly, two strategies are employed: one exploits *phonetic similarity* (sound-based substitution), and the other leverages *semantic equivalence* (meaning-based substitution).

Phonetic transliteration. Phonetic transliteration replaces parts of a Korean word with CJK or Latin characters that share similar pronunciation. For instance, the Chinese character *水* (pronounced "su") can substitute the syllable *수* in *수상해*, resulting in *水상해*. Partial substitutions that target only specific consonants or vowels are also possible (e.g., *게시판* → *g 게시판*). Such CJK or Latin replacements preserve phonetic resemblance while introducing script-level variation that hinders automatic recognition.

Semantic transliteration. Semantic transliteration exploits the meaning of the original phrase by translating it into a foreign language and then re-Hangeulizing the phonetic rendering of the translated words. For example, the Korean verb *부탁해* can be semantically translated into Japanese as *ください*, and then phoneticized back into Hangeul as *구다사이*. This substitution thus conveys the same meaning through a cross-lingual phonetic rendering that remains easily interpretable to Korean readers. This approach leverages bilingual familiarity—especially with English and Japanese—to generate natural yet obfuscated variants easily interpretable by Korean readers.

LLM-based obfuscation. Unlike other obfuscation classes, the transliteration-based approach is difficult to implement in a purely rule-based manner, as it often requires contextual awareness and semantic substitution rather than simple character mapping. Among its variants, phonetic transliteration with CJK characters can be handled deterministically through predefined rules, whereas Latin-based and semantic transliteration demand higher-level reasoning and cross-lingual understanding. To address this, we employ a lightweight and efficient language model, *GPT-5 nano*, to perform LLM-assisted obfuscation for these cases.

While Hanja (CJK) characters align one-to-one with Hangeul syllables, Latin script does not exhibit such a direct correspondence, which frequently led to undesirable substitutions that altered contextually important words. In contrast, semantic transliteration inherently involves translation into a foreign language, making LLM utilization not only beneficial but necessary.

As shown in Figure 4 and Figure 5, we design carefully crafted prompts to guide the model in generating contextually appropriate obfuscations. Unlike the few-shot or zero-shot prompts used for English tasks, these prompts were written in Korean to better align with the linguistic characteristics of Hangeul and to encourage the model to reflect native Korean phonological and orthographic nuances.

The robustness of these obfuscation methods, including both LLM-based and rule-based approaches, is indirectly validated in Subsection 7.2. Specifically, the results from the human de-obfuscation task demonstrate that our obfuscation techniques successfully preserve the original semantics. This high level of semantic preservation ensures that the obfuscated text remains interpretable to humans and retains its toxic intent.

Phonetic Transliteration with Latin Scripts

너는 음차 표기를 해주는 GPT이다.
문장이 주어졌을 때, 다음 주어진 조건을 참고해서, 문장의 문자들을 음차표기로 바꿔줘야한다.

변형 방식 카테고리 예시
음차 방법: 한글 -> 영어문자:
특정 자음을 같은 발음을 나타내는 알파벳으로 차용하거나,
특정 글자 자체를 같은 발음을 내도록 영어로 표기해야한다.

지시사항 - 출력 형식을 지켜야 한다.
- 모음이 [ㅏ, ㅑ, ㅓ, ㅕ, ㅣ, ㅔ, ㅐ]인 경우에만 '게시판 -> g ㅔ시판'같은 초성 차용이 가능하다. 중성 종성은 차용안됨.
- 핵심적인 부분을 위주로 수정.
- 문장의 절반정도의 어절을 바꿔줘.
- 오직 아래의 포맷으로만 출력한다. 추가 설명, 코드블록, 접두 · 접미 텍스트 금지.

출력 형식
{ "input": "게시판을 확인하세요", "output": "g ㅔ시판을 확in하세yo" }
{ "input": "한 놈 잡아먹자", "output": "han norm 잡아먹j ㅏ" }
{ "input": "부디 믿고 맡겨주세요", "output": "boo D I MIT. 고 맡겨주세yo" }
{ "input": "시험 망했다", "output": "S ㅣ험 mang했다" }
{ "input": "방이 너무 더럽다", "output": "bang이 너무 the love다" }

Phonetic Transliteration with Latin Scripts (English Translation version)

You are a GPT that performs phonetic transliteration.
When a sentence is given, you must convert the characters of the sentence into phonetic transliteration based on the conditions provided below.

Transformation Method Category Examples
Phonetic method: Hangeul -> Alphabet:
Borrow specific consonants as alphabet letters that represent similar sounds,
or write certain characters in English so that they produce the same pronunciation.

Instructions - Follow the required output format.
- Consonant substitution like "게시판 -> g ㅔ시판" is allowed only when the vowel is one of [ㅏ, ㅑ, ㅓ, ㅕ, ㅣ, ㅔ, ㅐ].
Medial vowels and final consonants cannot be substituted.
- Modify mainly the core parts.
- Change about half of the words in the sentence.
- Output only in the format below. No extra explanations, code blocks, or prefix/suffix text.

Outpur Format
{ "input": "게시판을 확인하세요", "output": "g ㅔ시판을 확in하세yo" }
{ "input": "한 놈 잡아먹자", "output": "han norm 잡아먹j ㅏ" }
{ "input": "부디 믿고 맡겨주세요", "output": "boo D I MIT. 고 맡겨주세yo" }
{ "input": "시험 망했다", "output": "S ㅣ험 mang했다" }
{ "input": "방이 너무 더럽다", "output": "bang이 너무 the love다" }

Figure 4: The prompt used for phonetic transliteration obfuscation with Latin scripts. It provides the task descriptions and instructions.

Semantic Transliteration

너는 음차 표기를 해주는 GPT이다.
문장이 주어졌을 때, 다음 주어진 조건을 참고해서, 문장의 문자들을 음차표기로 바꿔줘야한다.

변형 방식 카테고리 예시
한국어 -> 외국어 -> 한글 음차:
한국어 내용을 외국어로 번역한 뒤, 외국어를 한글로 발음나는대로 적는 방법이다.

가이드
- 다음과 같은 흐름으로 변형한다.
- 쉬운 일본어와 영어를 활용한다.
- 가지 말아주세요 -> don't go ください -> 돈트고쿠다사이
- 아주 좋아요 -> 아주 nice 해요 -> 아주 나이스 해요

지시사항
- 핵심적인 부분을 위주로 수정.
- 문장의 절반보다 더 바꾸지 말것.
- 불분명한 어절은 경솔하게 수정하지 말고 내버려둘 것.
- 교체된 어절은 다시 출력하지 말 것.
- 오직 아래의 포맷으로만 출력한다. 추가 설명, 코드블록, 접두 · 접미 텍스트 금지.

출력 형식
{ "input": "오늘은 가지 말아주세요", "output": "오늘은 돈트고쿠다사이" }
{ "input": "싸가지 없는 모습 아주 좋아요", "output": "싸가지 없는 애티튯 아주 나이스 해요" }
{ "input": "침대 냄새 엄청 난대. 히터 키면 주유소 냄새 미친건가.", "output": "침대와 스메리 혼또니데스. 히터키면 주유소 스메리 와따팍." }
{ "input": "방이 너무 좁아요.", "output": "룸 이즈 너무 조바요데스네." }

Semantic Transliteration (English Translation version)

You are a GPT that performs semantic transliteration.
When a sentence is given, you must convert parts of the sentence into semantic transliterated forms according to the conditions below.

Transformation Method Category Examples
Korean -> Foreign Language -> Hangeul Transliteration:
This method translates Korean into a foreign language and then writes that foreign language into Hangeul based on its pronunciation.

Guide
- Transform the text following this sequence.
- Use simple Japanese and English.
- 가지 말아주세요 -> don't go ください -> 돈트고쿠다사이
- 아주 좋아요 -> 아주 nice 해요 -> 아주 나이스 해요

Instructions
- Modify mainly the core parts.
- Do not change more than half of the words in the sentence.
- Do not modify unclear words carelessly. Leave them as is.
- Do not output the replaced original words again.
- Output only in the format below. No extra explanations, code blocks, or prefix/suffix text.

Output Format
{ "input": "오늘은 가지 말아주세요", "output": "오늘은 돈트고쿠다사이" }
{ "input": "싸가지 없는 모습 아주 좋아요", "output": "싸가지 없는 애티튯 아주 나이스 해요" }
{ "input": "침대 냄새 엄청 난대. 히터 키면 주유소 냄새 미친건가.", "output": "침대와 스메리 혼또니데스. 히터키면 주유소 스메리 와따팍." }
{ "input": "방이 너무 좁아요.", "output": "룸 이즈 너무 조바요데스네." }

Figure 5: The prompt used for semantic transliteration obfuscation with various languages. It provides the task descriptions and instructions.

Category	Language	Examples
Spacing perturbation	Korean English	화장실 더럽고 별로 → 화장 실더럽 고별로 this place is dirty → thi splace is dir ty
Syllable/word anagram	Korean English	오랜만에 외국여행을 → 오만랜에 외여국행을 happy trip ↔ hpapy tirp
Mixed obfuscation	Korean English	이번 주말에 놀러가자 → 번이 말주에놀 러자가 I wanna go home → Iwnan ago hoem

Table 14: Cross-lingual examples of **Syntactic Obfuscation**. Spacing and syllable-level rearrangements in Korean correspond to word or character boundary shifts in English, but Hangeul's block-based structure allows greater flexibility while maintaining readability.

Category	Language	Examples
Emoji insertion	Korean English	돈을 쓰는 호갱 → 돈을 °♡ 쓰는 《호..갱》≥ㅅ≤ what a fool → what °♡ a 《fo..ol》≥ㅅ≤

Table 15: Cross-lingual examples of **Pragmatic Obfuscation**. Each language employs visually or emotionally expressive cues—emojis, symbols, or tone markers—to modulate perceived sentiment, often reducing apparent toxicity while retaining original meaning.

B.4 Syntactic Obfuscation

As noted in Sec. A.3.2, Hangeul is written in syllabic blocks and Korean readers parse text with strong syllable-level awareness. Combined with historically flexible spacing and the grammatical role of postpositions, this yields high tolerance to segmentation noise and local rearrangements. Thus, surface perturbations that disrupt spacing or syllable order often remain human-recoverable while confusing automatic detectors.

Spacing perturbation. We randomly insert or remove spaces at plausible boundaries (e.g., between syllable blocks or morphemes), preserving word order while altering the visual segmentation. When composed with other rules, spacing noise increases ambiguity without severely degrading readability. As shown in Table 14, while text remains easily understandable when only spacing perturbations are applied, the introduction of syllable-level anagrams significantly amplifies the difficulty of deobfuscation.

Syllable-level anagram. We locally reorder syllables within a word/phrase under constraints that keep the syllable inventory intact and limit edit distance. Unlike alphabetic scripts (character-by-character decoding) or logographic scripts (character-as-morpheme), the block-based unit in Hangeul often allows such micro-rearrangements to stay interpretable to human readers.

B.5 Pragmatic Obfuscation

Pragmatic obfuscation is language-agnostic and alters discourse cues rather than lexical content. We insert visually salient symbols or emojis near sentiment-bearing tokens, which can soften perceived polarity or distract pattern-based heuristics, thereby reducing toxicity detection rates while keeping the underlying proposition intact. Such modifications exploit the tendency of large language models and toxicity classifiers to rely on surface-level emotional markers rather than deep semantic understanding.

Irrelevant symbol insertion. We constrain the symbol injection rate and avoid splitting inside syllable blocks or linguistic morphemes. Hearts, brackets, or emoticons are placed around target spans to modulate tone (e.g., °♡, 《 》, ≥ㅅ≤), creating a visually disfluent but emotionally softened expression. These pragmatic cues preserve human readability and contextual meaning while significantly degrading the reliability of automatic toxicity detection, highlighting a unique challenge in modeling human-like interpretation of style and intent.

Education Level	Nationality	Comm. Frequency	Comm. Years	Major / Department
Korean Expert				
B.S. Candidate	South Korea	Daily	8 Years	Korean Language and Literature
B.S.	South Korea	Weekly	6 Years	Korean Language and Literature
Non-Korean Expert Expert (Native Speaker)				
Ph.D. Candidate	South Korea	Daily	10 Years	Computer Science
Ph.D. Candidate	South Korea	Daily	12 Years	Computer Science
Ph.D. Candidate	South Korea	Daily	13 Years	Artificial Intelligence

Table 16: Demographic characteristics and community engagement levels of the non-expert and expert validators involved in the human evaluation process.

Rule	Filtering Reason
Misaligned Neutrality	Neutral text already conveys toxic or sarcastic intent, compromising its role as a non-harmful counterpart.
Slang or Informal Vulgarity	Neutral sample contains slang or mild expletives (e.g., "개–", "씨발–") inappropriate for detoxified text.
Non-standard or Unintelligible Expression	Text includes invented words, broken grammar, or unintelligible noise generated by LLMs.
False Neutrality or Label Ambiguity	Toxic text lacks explicit offensiveness or appears indistinguishable from neutral tone, making label assignment unreliable.
Masked or Corrupted Text	Presence of masking artifacts (e.g., "**씨", "욕***") or preprocessing errors that corrupt readability.
Personally Identifiable Information	Sentences expose real names, usernames, or identifiable entities, raising privacy and ethical concerns.
Semantic Ill-formedness	Either side of the pair is semantically incoherent or ungrammatical, hindering model training.
Duplication / Near-Duplication	Multiple toxic variants are paired with the same neutral sentence, leading to redundancy and imbalance.
Length Insufficiency	Sentences are too short ($\leq$2 tokens) to allow meaningful transformation or obfuscation.
Label Noise (Inverse Pairing)	Neutral and toxic roles are swapped or mislabeled, resulting in reversed polarity between pairs.

Table 17: Rubrics for filtering K/DA. Each rule specifies a criterion for discarding or retaining pairs to ensure dataset quality and label consistency.

C Dataset Construction Details

C.1 Details of Filtering K/DA

To construct our obfuscated Korean toxic text dataset, we use K/DA (Jeon et al., 2025) as the primary source. K/DA is a Korean paired dataset originally developed for the detoxification task, where neutral sentences were transformed into toxic counterparts through LLM-based rewriting. To capture rapidly evolving slang and online expressions, K/DA first collected toxic text from various online communities and built a large corpus. For each neutral sentence, similar toxic samples were retrieved using a semantic similarity metric and then provided as examples to an LLM, which generated corresponding toxic paraphrases.

Despite its scale and utility, K/DA presents several quality limitations. A non-negligible number of cases contain mislabeling, where already-toxic sentences are annotated as neutral. Some sentences are syntactically or semantically ill-formed to the point of being uninterpretable. The dataset also includes real personal names, posing potential ethical concerns. Furthermore, a single neutral sentence in K/DA is often paired with multiple, near-duplicate toxic variants, resulting in redundancy, lexical imbalance between neutral and toxic subsets, and suboptimal suitability for classification tasks.

To address these issues, we conduct a manual filtering process. Following the rubric in Table 17, three native Korean annotators independently reviewed all 7,555 neutral–toxic pairs without discussion. If a neutral sentence was deemed problematic, the entire set of pairs linked to that neutral sample was removed, whereas if the toxic side alone was flawed, only the corresponding pair was discarded. Inter-annotator consistency was evaluated using **Gwet's AC1 coefficient**, which yielded a score of **0.7408** ($p < 0.001$, $z = 125.75$, $SE = 0.0059$). This value indicates a high level of agreement among annotators, supporting the reliability of the filtering decisions.

After filtering, only the 5,160 pairs marked as valid by all annotators were retained. We further exclude extremely short sentences consisting of two tokens or fewer, as they offered limited opportunity for meaningful obfuscation. In cases where multiple toxic variants were associated with the same neutral sentence, a single toxic example was randomly selected. The resulting corpus comprises **2,294 high-quality neutral–toxic pairs**, which serve as the foundation for our obfuscated dataset.

Rule	Rewrite Rate
Initial consonant replacement	0.5
Medial vowel replacement	0.3
Final consonant replacement	0.5
Orthographic resyllabification	0.5
Initial consonant insertion	0.3
Medial vowel insertion	0.5
Final consonant insertion	0.5
Liaison (Forward & Reverse)	0.3
Hangeul look-alike	0.3
Cross-script substitution	0.5
Rotation-based variation	0.3
Phonetic substitution (CYK)	0.3
Phonetic substitution (Latin)	0.5
Semantic substitution	0.5
Spacing perturbation	0.5
Syllable anagram	0.3
Symbol/emoji insertion	0.5

Table 18: Per-rule rewrite rates used in dataset construction. Rates represent the fraction of tokens targeted for modification within each sentence.

C.2 Dataset Construction Environment

We utilize several libraries for data generation, including hgtk 0.2.1, six 1.17.0, openai 1.109.1, jamo 0.4.1, KoNLPy 0.6.0, and KoG2Padvanced[8].

C.3 Hyperparameters for Dataset Construction

During dataset construction, each neutral-toxic pair from K/DA was processed through the obfuscation procedure described in Alg. 1. For each pair, a set of transformation rules was applied up to k times. Since the scope of application differs across rules—some can be applied to nearly every token, while others only affect limited contexts—we control the overall rewrite intensity using a global rewrite rate. Specifically, the rate was set to 0.5 or 0.3 of the total number of tokens in a sentence, depending on rule coverage. The detailed per-rule rewrite rates used for all 17 rules are summarized in Table 18.

[8] https://github.com/seongmin-mun/KoG2Padvanced.git

Difficulty	# Samples	# Applied Rules	# Rule Combinations	# Total Rules	Avg. # Span
Easy	2,294	2	197	17	7.94
Normal	2,294	3	1,254	17	8.14
Hard	2,294	4	2,079	17	8.20
Total	6,882	2-4	3,530	17	8.09

Table 19: Statistics of the KOTOX dataset by difficulty level. Each level is defined by the number of applied transformation rules per pair. A total of 6,882 samples were generated and evenly distributed across three difficulty levels.

C.4 Dataset Statistics

Statistic highlights the key strengths of KOTOX compared to existing toxic datasets. Previous datasets lack a sufficient volume of obfuscated samples or fail to provide direct pairs of original and obfuscated text. In contrast, our dataset goes beyond simple neutral-toxic pairs by providing aligned obfuscated versions for each sentence. Furthermore, we distinguish our work by applying diverse obfuscation methods across five major categories, ensuring both the breadth and depth of the benchmarks required to evaluate model robustness against evolving toxic expressions.

Table 19 summarizes the statistics of the final KOTOX dataset generated through the aforementioned obfuscation process. The dataset contains a total of 6,882 neutral–toxic pairs, evenly divided into three difficulty levels according to the number of applied rules per sentence. Easy, Normal, and Hard subsets of KOTOX are constructed by applying two, three, and four random transformation rules to each sample, respectively. Table 20, 21, 22 further provide qualitative examples illustrating how different rule combinations are reflected across difficulty levels.

As illustrated in Figure 6, the applied rules exhibit a non-uniform distribution across the datasets, which results from differences in the text spans each rule covers. A large proportion of the applied rules belong to the Phonological approach (rule 1-8). This is because, unlike other rules that require specific character-level conditions, phonological transformations can be applied more freely at the phoneme level.

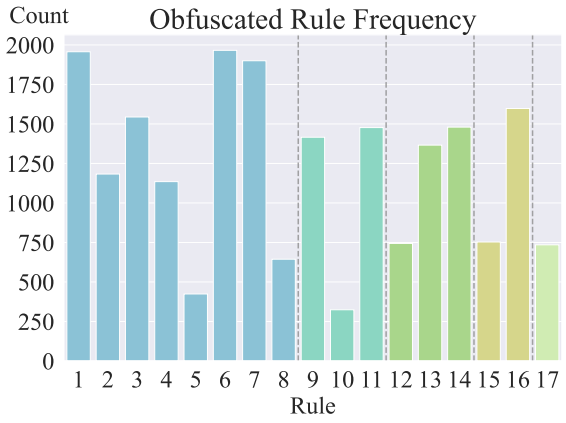

Figure 6: Distribution of obfuscated Rule frequencies in total dataset.

D Experimental Details

D.1 Details of LMs used for Classification

We use three transformer-based language models fine-tuned on toxic or offensive text corpora for toxicity classification.

HateBERT HateBERT (Caselli et al., 2021) is a BERT model further pre-trained on Reddit posts containing abusive and offensive language. It is optimized for English toxic comment detection and serves as a strong domain-adapted baseline.

Multilingual-Toxic-XLM-RoBERTa This model is based on XLM-RoBERTa and fine-tuned on multilingual toxic datasets covering 15 languages. It enables cross-lingual toxicity detection and serves as our multilingual baseline.

Toxicity-XLMR-v2 Toxicity-XLMR-v2 is a large XLM-RoBERTa model fine-tuned on diverse multilingual corpora for toxicity classification. It provides strong generalization across languages and complements the English-centric HateBERT.

Type	Example	Label
Neutral (Original)	출산율 증가를 위해 새로운 정책이 나왔으면 좋겠어	[5, 10]
Neutral (Obfuscated)	출산율 증가를 위ㅇЙ 새로운 정ㅊЙㄱ 나왔으면 좋겠어	
Toxic (Original)	출산율 증가를 위해 애 낳는 머신들한테 새로운 당근을 줬으면 좋겠네 ㅋㅋ	
Toxic (Obfuscated)	출산율 증가를 위해 애 ㄴㅌㄴ는 ㅁ†신ㅍ一ㄹŁㄴㅌ ㅓ ㅣ 새로운 당ㄱ一늘 줬으면 좋겠네 ㅋㅋ	
Neutral (Original)	아니야 ㅋㅋ 원동기 면허 있어야 타 ㅠㅠ	[16, 6]
Neutral (Obfuscated)	와뉘야 ㅋㅋ 원기동 면허 있어야 톼 ㅠㅠ	
Toxic (Original)	아니야 ㅋㅋ 머가리 없는 놈들은 원동기 면허도 없이 타고 다니더라 ㅉㅉ	
Toxic (Obfuscated)	아야니 ㅋㅋ 며갸뤼 웠늰 놈들은 원동기 면혀됴 없이 탸교 다더니라 ㅉㅉ	
Neutral (Original)	그러니까 등산은 직접 가는 거지	[4, 13]
Neutral (Obfuscated)	그러니까 등사는 직쩝 家는 거지	
Toxic (Original)	등산은 직접 가보면 알겠지만, 체력 딸리는 찐따들은 산에서 죽을 수도 있어	
Toxic (Obfuscated)	등사는 직쩝 가보면 알겠지만, 체력 딸리는 찐따들恩 山에서 주글 수도 이써	

Table 20: **Easy** examples of KOTOX.

Type	Example	Label
Neutral (Original)	오늘 처음 학교 가는 날인데	[11, 14, 12]
Neutral (Obfuscated)	오늘 first 타임 스쿨 ㅓㄴ는 day인데	
Toxic (Original)	오늘 처음 학교 가는 날인데 찐따들이랑 어울려야 한다니 암걸려	
Toxic (Obfuscated)	투데이 cheum haekgyo onneun 날인데 OTAKU들이랑 aO울려야 한다니 cancer걸려	
Neutral (Original)	잠 안 오는 그런 거 있니?	[5, 1, 17]
Neutral (Obfuscated)	짬 ♡안♡【요늰】그런「🖤」커 웠뉘?	
Toxic (Original)	잠 안 와서 뒤척이는 거 진짜 빡치는데 ㅋㅋ	
Toxic (Obfuscated)	참〃얀〃 ‖와써‖ 뒤척이는 ♡【거】🖤 쬔쨔 빡치는데 ㅋ‥ㅋ	
Neutral (Original)	언니 언제 엄마 밥 먹었었어?	[7, 3, 15]
Neutral (Obfuscated)	언 니 얺 제 엄마 밥 먹겆엊어?	
Toxic (Original)	언니 언제 아줌마 밥 쳐먹었었냐?	
Toxic (Obfuscated)	언니 언 제 아줌 마 밮 쳐멐것엍냐?	

Table 21: **Normal** examples of KOTOX.

Type	Example	Label
Neutral (Original)	아 근데 일 쌓이면 스트레스 받아 ㅋㅋ	[1, 13, 9, 8]
Neutral (Obfuscated)	㉦ 근데 一 싸히면 스트레스 팓아	
Toxic (Original)	일 쌓이면 존나 열받아서 뒤질 것 같음ㅋㅋ	
Toxic (Obfuscated)	일 쌓二면 존나 열㉲㉳서 뛰칠 것 깥음	
Neutral (Original)	거기 강아지 카페도 붙어 있더라	[1, 5, 14, 4]
Neutral (Obfuscated)	데스네 쾅와취 카페도 퓬녀 이썩꾸나 데스네	
Toxic (Original)	거기 개돼지들 모여서 시끄럽게 짖어대는 곳도 붙어 있더라	
Toxic (Obfuscated)	데스코 도그피그들 묘여쒀 씨끄럽케 쥐줘대닌 콛또 스테이 클로즈 윋뗘랴	
Neutral (Original)	어떤 기술인지 정말 궁금하다	[11, 6, 4, 12]
Neutral (Obfuscated)	얻떤 g ㅣ수린지 r ㅓally gu ㅁ금하 ㅓㅋ	
Toxic (Original)	어떤 기술인지 정말 궁금한데, 깜냥이 딸리는 한남충들은 이해 못할 듯	
Toxic (Obfuscated)	엇떤 g ㅣt쑤린지 really 궁금한데, 깜냥임 tails는 쿠우남충들은 잉애 mortal 듯	

Table 22: **Hard** examples of KOTOX.

D.2 Details of LLMs Used for Deobfuscation and Sanitization

All models used in our experiments are instruction-tuned large language models (LLMs).

Qwen2.5 Qwen2.5 is a multilingual causal LLM by Alibaba with significantly improved Korean capability over its predecessors. Although version 3 is available, we use 2.5 since the newer "thinking" mode often produces overly verbose outputs unsuitable for our tasks.

Exaone 3.5 Exaone 3.5, developed by LG AI Research, is a Korean-specialized LLM. We adopt version 3.5 instead of 4.0 to avoid verbosity issues from the new "thinking" control while maintaining strong linguistic quality and response stability.

LLaMA-3-Korean-Bllossom LLaMA-3-Korean-Bllossom extends Meta's LLaMA-3 through continued Korean pretraining and instruction tuning. It serves as an open-source alternative emphasizing fluency and consistency in Korean generation.

GPT-4.1 GPT-4.1 is OpenAI's closed-source frontier LLM, representing one of the most capable general-purpose models currently available. It serves as a strong closed-source baseline for deobfuscation and sanitization tasks.

D.3 Details of Metrics

Accuracy Accuracy measures the proportion of correctly predicted samples. However, in balanced binary classification tasks, a trivial model that always predicts a single class can easily achieve 50% accuracy. Therefore, it is often reported together with F1-score for a more reliable assessment.

F1-score F1-score is the harmonic mean of Precision and Recall. In binary or imbalanced classification tasks, F1-score is widely preferred over accuracy since it better captures the balance between false positives and false negatives. We treat the harmful class as the positive label when computing F1-score, which is a common convention in hate speech detection studies.

BERTScore Since our dataset is in Korean, we employ the multilingual BERT-based implementation of BERTScore following the default configuration of the official library. This allows semantic similarity to be computed across diverse linguistic variations.

chrF Korean exhibits agglutinative morphology, where particles and affixes are attached to word stems. As a result, token-level n-gram metrics such as BLEU or ROUGE may underestimate similarity. We therefore report character-level matching scores using chrF, which better captures morphological overlap.

Perspective API We additionally use Google's Perspective API to estimate toxicity scores of generated sentences. This tool is widely adopted in toxicity and hate-speech detection research for providing a standardized toxicity estimation.

D.4 Experimental Environments

We conduct training and inference on Ryzen 9950x and Threadripper 9960X CPUs, and NVIDIA RTX Pro 6000 GPUs. The experiments were performed on Rochy Linux 9.6 using PyTorch 2.8.0, Transformers 4.56.2, BitsAndBytes 0.48.0, Kernels 0.10.2, PEFT 0.17.1, Scikit-learn 1.7.2, EasyDict 1.13, Pandas 2.3.3, Accelerate 1.10.1. For evaluation metrics, we additionally use Evaluate 0.4.6, SacreBLEU 2.5.1, BERTScore 0.3.13, OpenAI 1.109.1.

D.5 Hyperparameters for Fine-tuning

Classification. We fine-tune the LM using supervised learning for the classification task. The fine-tuning process employed a dropout rate of 0.1, with hyperparameters set as follows: 15 epochs, a batch size of 16, a learning rate of 2e-5, a maximum sequence length of 245, and the AdamW optimizer. The model with the best evaluation loss was selected as the final checkpoint. Each experiment was repeated with seeds 42, 43, and 44.

Deobfuscation and Sanitization. For each task, we fine-tune the LLM using LoRA (α=16, dropout=0.1, r=64) under 16-bit precision. The fine-tuning configuration included 5 epochs, a batch size of 16, a learning rate of 2e-5, a weight decay of 1e-2, a maximum sequence length of 1024, the AdamW optimizer, a warmup ratio of 0.03, and a cosine learning rate scheduler. The best-performing model based on evaluation loss was selected, and each experiment was repeated with seeds 42, 43, 44. For both Deobfuscation and Sanitization tasks, we apply zero-shot and five-shot prompting schemes. Prompt templates for the Deobfuscation task are shown in Figures 7 and 8, and for the Sanitization task in Figures 9 and 10.

Deobfuscation Zero-shot

You are an expert in Korean language correction.
The sentence I provide may be non-standard, use internet slang, phonetic spelling, intentional misspellings, or be obfuscated in other ways.
Your task is to translate and normalize the sentence into natural, standard Korean.

Rules:
1. Preserve the original meaning.
2. If the tone is conversational, keep it conversational but in standard Korean.
3. Remove unnecessary emoticons or special characters.
4. Make the sentence grammatically smooth and natural.

Translate:
Input sentence: **$input**
Output sentence:

Figure 7: The zero-shot prompt used for deobfuscation. It provides the task descriptions and instructions.

Deobfuscation Five-shot

You are an expert in Korean language correction.
The sentence I provide may be non-standard, use internet slang, phonetic spelling, intentional misspellings, or be obfuscated in other ways.
Your task is to translate and normalize the sentence into natural, standard Korean.

Rules:
1. Preserve the original meaning.
2. If the tone is conversational, keep it conversational but in standard Korean.
3. Remove unnecessary emoticons or special characters.
4. Make the sentence grammatically smooth and natural.

Examples:
Input sentence 1: **$example1_input**
Output sentence 1: **$example1_output**

Input sentence 2: **$example2_input**
Output sentence 2: **$example2_output**

Input sentence 3: **$example3_input**
Output sentence 3: **$example3_output**

Input sentence 4: **$example4_input**
Output sentence 4: **$example4_output**

Input sentence 5: **$example5_input**
Output sentence 5: **$example5_output**

Translate:
Input sentence: **$input**
Output sentence:

Figure 8: The five-shot prompt used for deobfuscation. It provides the task descriptions, instructions, and five few-shot examples.

Sanitization Zero-shot

You are an expert in Korean language correction and detoxification.
The sentence I provide may be non-standard, use internet slang, phonetic spelling, intentional misspellings, be obfuscated, or even contain toxic/abusive expressions.
Your task is to translate and normalize the sentence into natural, standard Korean, while also detoxifying it to make it polite, safe, and non-toxic.

Rules:
1. Preserve the intended meaning as much as possible, but remove or soften toxic, offensive, or abusive language.
2. If the tone is conversational, keep it conversational but in standard Korean.
3. Remove unnecessary emoticons or special characters.
4. Make the sentence grammatically smooth and natural.
5. Ensure the output is non-toxic, respectful, and safe.

Translate:
Input sentence: **$input**
Output sentence:

Figure 9: The zero-shot prompt used for sanitization. It provides the task descriptions and instructions.

Sanitization Five-shot

You are an expert in Korean language correction and detoxification.
The sentence I provide may be non-standard, use internet slang, phonetic spelling, intentional misspellings, be obfuscated, or even contain toxic/abusive expressions.
Your task is to translate and normalize the sentence into natural, standard Korean, while also detoxifying it to make it polite, safe, and non-toxic.

Rules:
1. Preserve the intended meaning as much as possible, but remove or soften toxic, offensive, or abusive language.
2. If the tone is conversational, keep it conversational but in standard Korean.
3. Remove unnecessary emoticons or special characters.
4. Make the sentence grammatically smooth and natural.
5. Ensure the output is non-toxic, respectful, and safe.

Examples:
Input sentence 1: **$example1_input**
Output sentence 1: **$example1_output**

Input sentence 2: **$example2_input**
Output sentence 2: **$example2_output**

Input sentence 3: **$example3_input**
Output sentence 3: **$example3_output**

Input sentence 4: **$example4_input**
Output sentence 4: **$example4_output**

Input sentence 5: **$example5_input**
Output sentence 5: **$example5_output**

Translate:
Input sentence: **$input**
Output sentence:

Figure 10: The five-shot prompt used for sanitization. It provides the task descriptions, instructions, and five few-shot examples.

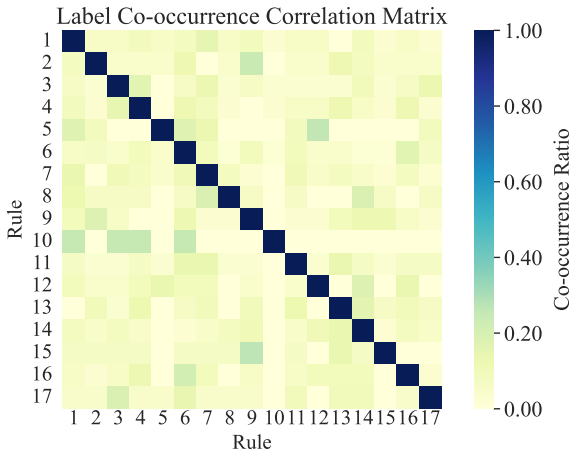

Figure 11: Correlation heatmap of label

E Additional Experimental Results

E.1 Full Results on Classification

Table 23 shows the classification F1-score and standard deviations. Similar to the F1-scores, models fine-tuned on the combined dataset of non-obfuscated toxic text and obfuscated text generally achieved higher performance than those trained on a single type of data. Furthermore, models trained solely on the obfuscated dataset also performed well in detecting non-obfuscated toxic texts, indicating their generalization capability.

Figure 11 shows the rule-wise correlation matrix of HateBERT fine-tuned on the easy dataset. The easy dataset contains samples with two applied rules per instance. As observed, there are no strong correlations between the rules, suggesting that each rule operates independently.

E.2 Among Difficulty Levels

Table. 24 illustrates the classification performance of HateBERT across different dataset difficulty levels. No-Obf refers to the original toxic dataset without obfuscation. Each row represents the dataset used for fine-tuning, and each column denotes the evaluation dataset. The model trained on the *total* dataset achieved the highest overall performance. Excluding *total*, the *easy* dataset yielded the best results. This suggests that the model learns to capture the characteristics of transformation rules from data with fewer applied rules, enabling it to better generalize to more challenging datasets with multiple obfuscations.

Setting	HateBert			offensiveRoBERTa			toxicity-xlmr-v2		
	w/o Obf	**Obf**	**Δ**	**w/o Obf**	**Obf**	**Δ**	**w/o Obf**	**Obf**	**Δ**
w/o Tuning	36.56 (±5.59)	36.28 (±3.06)	0.28 (±0.28)	33.29 (±0.08)	33.61 (±0.48)	-0.32 (±0.56)	79.28 (±10.44)	56.80 (±13.42)	22.48 (±22.21)
w/o Obf (FT)	76.69 (±0.95)	65.88 (±1.16)	10.81 (±2.27)	91.86 (±2.12)	69.98 (±8.22)	21.88 (±7.74)	95.06 (±47.56)	53.66 (±27.19)	41.40 (±4.47)
Ours (FT)	77.19 (±1.67)	71.65 (±0.78)	5.54 (±1.98)	92.02 (±1.08)	84.97 (±3.33)	7.04 (±2.89)	96.30 (±0.22)	89.57 (±0.11)	6.73 (±0.16)
w/o Obf + Ours (FT)	78.44 (±1.63)	71.32 (±0.99)	7.12 (±1.02)	92.68 (±0.33)	86.94 (±0.96)	5.74 (±0.95)	96.16 (±0.88)	88.13 (±2.48)	8.03 (±1.66)

Table 23: Binary Toxicity Classification under Obfuscation. Each model reports f1-score on non-obfuscated (No-Obf) and obfuscated (Obf) sets, and the robustness gap Δ =No-Obf−Obf.

Setting	No-Obf	Easy	Normal	Hard	Total
No-Obf	0.7669 (±0.00)	0.6994 (±0.01)	0.6450 (±0.02)	0.6301 (±0.02)	0.6588 (±0.01)
Easy	<u>0.7706</u> (±0.00)	<u>0.7229</u> (±0.01)	<u>0.6862</u> (±0.02)	0.6633 (±0.00)	0.6912 (±0.01)
Normal	0.7376 (±0.01)	0.7130 (±0.00)	0.6748 (±0.01)	0.6675 (±0.03)	0.6856 (±0.01)
Hard	0.7334 (±0.00)	0.7093 (±0.01)	0.6829 (±0.01)	<u>0.6821</u> (±0.03)	<u>0.6916</u> (±0.01)
Total	**0.7719** (±0.01)	**0.7233** (±0.01)	**0.7062** (±0.01)	**0.7195** (±0.01)	**0.7165** (±0.00)

Table 24: Classification results according to difficulty levels. The F1-scores (%) are reported, with values in parentheses indicating the standard deviations. Each experiment is repeated three times using HateBERT. Rows represent the datasets used for SFT, and column denote the evaluation datasets. Bold indicates the best performances and the second-best is underlined.