

Breaking the Cloak! Unveiling Chinese Cloaked Toxicity with Homophone Graph and Toxic Lexicon

Xuchen Ma¹, Jianxiang Yu¹, Wenming Shao², Bo Pang², Xiang Li^{1*}

¹School of Data Science and Engineering, East China Normal University

²Shanghai EastWonder Info-tech Co., Ltd.

{xuchenma, jianxiangyu}@stu.ecnu.edu.cn

{simon, pangbo}@wdit.com.cn

xiangli@dase.ecnu.edu.cn

Abstract

Social media platforms have experienced a significant rise in toxic content, including abusive language and discriminatory remarks, presenting growing challenges for content moderation. Some users evade censorship by deliberately disguising toxic words through homophonic cloak, which necessitates the task of unveiling cloaked toxicity. Existing methods are mostly designed for English texts, while Chinese cloaked toxicity unveiling has not been solved yet. To tackle the issue, we propose C²TU, a novel training-free and prompt-free method for Chinese cloaked toxic content unveiling. It first employs substring matching to identify candidate toxic words based on Chinese homophone graph and toxic lexicon. Then it filters those candidates that are non-toxic and corrects cloaks to be their corresponding toxicities. Specifically, we develop two model variants for filtering, which are based on BERT and LLMs, respectively. For LLMs, we address the auto-regressive limitation in computing word occurrence probability and utilize the full semantic contexts of a text sequence to reveal cloaked toxic words. Extensive experiments demonstrate that C²TU can achieve superior performance on two Chinese toxic datasets. In particular, our method outperforms the best competitor by up to 71% on the F1 score and 35% on accuracy, respectively. Our code and data are available at <https://github.com/XDxc-cuber/C2TU-Chinese-cloaked-toxicity-unveiling>.

Disclaimer: *The paper contains content that may be profane, vulgar, or offensive.*

1 Introduction

With the exponential growth of user-generated contents, online social media platforms have become an inevitable communication tool with massive users (Omipidan and Sanusi, 2024; Bryikhanova

et al., 2021; Bala, 2014). Although the rapid information dissemination has clear benefits, a significant amount of toxic contents have emerged on social platforms over the past decade, such as abuse, discrimination, and cyberbullying (Sheth et al., 2022; Recuero, 2024; Kim et al., 2021; Hanscom et al., 2024). Social networks are thus facing increasingly severe challenges in governing toxicity.

Toxicity detection has recently attracted extensive attention (Anjum and Katarya, 2024; Lima et al., 2024; Schmidhuber and Kruschwitz, 2024), but most of the proposed methods (Phute et al., 2023; Zhang et al., 2024; Deng et al., 2022; Zhao et al., 2025) can only output binary predictions rather than recover the original toxic expressions hidden in cloaked text. Further, to evade censorship, some users on social platforms intentionally disguise toxic words by replacing parts or all of the words with homophonic characters or emojis (Xiao et al., 2024; Zhang et al., 2024; Kirk et al., 2022). For example, replace “操” (cào, means “f*ck”) with “草” (cǎo, means “grass”), or “垃圾” (lā jī, means “rubbish”) with “辣鸡” (là jī, means “spicy chicken”) in Chinese. These deliberate cloaks significantly degrade the effectiveness of existing toxicity detection methods (Xiao et al., 2024) and also necessitate the task of *unveiling cloaked toxicity*, i.e., restoring cloaked toxic words to their original proto-toxic words. While there have been some methods (Gong et al., 2019; Dong et al., 2023) proposed to solve the task, most of them are specially designed for English texts, which cannot be directly used for Chinese due to the different characteristics of the two languages. Further, although a recent work (Xiao et al., 2024) points out the existence of Chinese cloak toxic content, they fail to offer a solution. Therefore, to bridge the gap, a research question naturally arises: *Can we develop a model to unveil Chinese cloaked toxic contents?*

We notice that Chinese Spelling Correction (CSC), which aims to correct misspelled Chinese

*Corresponding Author

characters, shares some similarities with our task. However, the two tasks are not entirely equivalent. While CSC deals with unintentional user typos, misspellings in toxic content are often deliberate. Moreover, the corpora used in CSC tasks typically consist of more standardized language, which differs significantly in distribution from the toxicity found on the internet. Therefore, although models like SCOPE (Li et al., 2022) and Simple-CSC (Zhou et al., 2024) have demonstrated strong performance on CSC tasks, directly applying them to this restoration task may not yield satisfactory results (see Section 4).

In this paper, we study Chinese Cloaked Toxicity Unveiling as a sequence restoration task, and propose the C²TU model (see Figure 1). Specifically, we utilize a homophone graph and a toxic lexicon to identify potential toxic words within the input text, thereby transforming the problem into a candidate toxic words filtering task. This task aligns well with BERT model’s pretraining objective of masked language modeling. Hence, we first employ BERT model to compute the probability difference between tokens derived from raw and toxic words at specific positions in a sentence. If toxic tokens have larger probabilities, we then unveil the cloak. After that, we further explore using LLMs to address the filtering task. Similar to the BERT-based approach, we attempt to calculate word occurrence probabilities to decide whether a replacement should occur. To better exploit full sentence context, we reformulate the filtering score as a sentence-level likelihood difference. Since LLMs are well-suited for computing sentence-level likelihoods, this formulation allows us to incorporate the full semantic context of a sentence. Finally, we summarize the main contributions of our paper as:

- We formulate Chinese cloaked toxicity unveiling as a sequence restoration task and propose a training-free and prompt-free method C²TU for Chinese cloaked toxicity unveiling. Different from most existing methods for English and from generic toxicity detection, to our best knowledge, we are the first to solve the problem under Chinese.
- We propose BERT- and LLM-based filtering modules for candidate toxic words. In particular, the LLM variant uses a sentence-level likelihood formulation so it can exploit the full semantic context during filtering.
- We conduct extensive experiments to evaluate the model performance on two Chinese toxic datasets. The results show that our methods are more competitive than other baselines w.r.t. both F1 score and accuracy metrics. In particular, our method outperforms the best competitor by up to 71% on the F1 score and 35% on accuracy.

2 Related Work

2.1 Cloaked Toxicity Unveiling

While extensive researches have been conducted on toxic content detection (Davidson et al., 2017; Jafari et al., 2024; Arora et al., 2023; Ghosh et al., 2024; Phute et al., 2023; Zhang et al., 2024; Deng et al., 2022; Lu et al., 2023), most of them are not specially designed for cloaked toxic words (Xiao et al., 2024). Further, the reveal of deliberately cloaked toxic words emerges as a critical research challenge worthy of attention.

Some recent efforts have been made to address the issue by handling intentional obfuscation through character scrambling and misspellings in English. For example, some works (Gong et al., 2019) leverage contextual information to correct intentionally misspelled words, while others (Dong et al., 2023) focus on addressing word order disruptions by employing LLMs for robust comprehension of noisy text. However, these methods are designed for English. Unlike English—a word-based language where obfuscation typically alters intra-word characters—Chinese operates at the character level but cloaks content at the word level. This fundamental linguistic disparity prevents direct application of existing English-based methods to Chinese. However, research on unveiling Chinese cloaked toxic content remains largely unexplored.

Recent Chinese toxicity studies have also expanded the problem setting with multimodal benchmarks, span-level target-aware extraction, and homophonic-perturbation detection (Yang et al., 2025; Bai et al., 2025; Wu et al., 2025). These works strengthen detection and extraction resources, but they still stop short of restoring cloaked spans to their original toxic expressions.

2.2 Chinese Spelling Correction

Our task shares similarities with the CSC task, which aims to identify and correct Chinese spelling errors based on contextual, phonetic and graphemic information. Some existing methods are based on

small language models (e.g., BERT) (Zhang et al., 2020; Cheng et al., 2020; Li et al., 2022). For example, SCOPE (Li et al., 2022) utilize phonetic knowledge of Chinese characters in conjunction with BERT for spelling correction. With the advent of LLMs, there are also LLM-based models (Dong et al., 2024; Zhou et al., 2024; Li et al., 2024). A representative model Simple-CSC (Zhou et al., 2024) is a training-free and prompt-free method, treating LLMs purely as language models to perform spelling correction at the token level.

However, the CSC task is not well aligned with our task, as the substitution of toxic words is often intentional, and toxic texts themselves are abnormal texts that deviate from regular language usage. Therefore, we pay attention to Chinese cloaked toxicity unveiling in this paper.

3 Methodology

3.1 Graph and Lexicon Construction

3.1.1 Chinese Homo-graph

Given a Chinese toxic speech dataset, we first construct a homophone graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ to capture the likely cloak between tokens, where each node in $\mathcal{N}$ represents a Chinese character, and each edge in $\mathcal{E}$ indicates the phonetically similar relation. Specifically, we extract all Chinese characters from the dataset as nodes, then use the open-source library pypinyin¹ to obtain their pinyin regardless of tones. Then characters with identical pinyin are connected. Each node also has a self-loop, as each character is considered to have a homophonic relation with itself. We further consider polyphonic characters and phonetically similar pronunciations based on regional dialects.

Polyphonic Characters: Some Chinese characters have multiple pronunciations. If a character has at least one pronunciation matching another character’s, an edge is linked between them.

Dialectal Relationships: There also exist dialectal phonetic confusions in Chinese, where users may replace characters based on their dialect pronunciations. Common confusions include:

- Retroflex and non-retroflex sounds (Dong et al., 2006), such as “山” (shān, means “mountain”) and “三” (sān, means “three”).
- Front and back nasal sounds (Chen and Guion-Anderson, 2011), such as “应” (yīng, means “should”) and “因” (yīn, means “reason”).

- Initial consonants “n” and “l” (Zhou and Wu, 2019; Koffi, 2019), such as “男” (nán, means “male”) and “蓝” (lán, means “blue”).

Considering these confusions, we introduce five types of additional relations: “n” $\leftrightarrow$ “l”, “zh” $\leftrightarrow$ “z”, “ch” $\leftrightarrow$ “c”, “sh” $\leftrightarrow$ “s” and “*ng” $\leftrightarrow$ “*n”. For example, “男” (nán) and “蓝” (lán) share an “n” $\leftrightarrow$ “l” confusion, so an edge is added between node “男” and node “蓝”, although their pinyin representations are not entirely identical.

3.1.2 Toxic Lexicon

We also utilize a toxic lexicon to introduce external knowledge about toxicity. The lexicon is a set of toxic words denoted as $\mathcal{L} = \{l_1, \dots, l_m\}$. While we notice that some public datasets (e.g., ToxiCN (Lu et al., 2023)) have their own toxic lexicon released, they also contain many toxic words with homophone cloak. As these cloaks are frequently and commonly used on social platforms, they gradually become substitutes of their corresponding toxic contents. We take “垃圾” (lā jī, means “rubbish”) and “辣鸡” (là jī, means “spicy chicken”) as an example. Here, the proto-toxic word is “垃圾”, while “辣鸡” is a cloaked substitution commonly used in social media, even more frequently than the proto-toxic word. These cloaks should also be removed instead of remaining in the vocabulary as toxic knowledge. Therefore, to keep a simple and clean toxic lexicon, we only retain proto-toxic words.

3.2 Toxic Word Matching

Given a text sequence X , we enumerate substring w in X and match each w against toxic words $l \in \mathcal{L}$ using the homophone graph $\mathcal{G}$. Formally, a substring w of length N is considered as a candidate toxic word if and only if $\exists l \in \mathcal{L}, \text{len}(w) = \text{len}(l)$ and $\forall k \in \{1, 2, \dots, N\}, \mathcal{G}.\text{HasEdge}(w_k, l_k) = 1$, where w_k is k -th Chinese character in w , and so does l_k . Here, $\text{len}(\cdot)$ is the length of the input string. The function $\mathcal{G}.\text{HasEdge}(\cdot, \cdot) = 1$ indicates that two nodes are linked in $\mathcal{G}$.

The above matching method could have a small *false negative rate*, i.e., any homophonic substitution will be unveiled given a comprehensive homophone graph $\mathcal{G}$ and lexicon $\mathcal{L}$. However, it may also yield a high *false positive rate* by incorrectly matching non-toxic word as toxic word. For example, “干净” (gān jìng, means “clean”) will be matched to “杠精” (gàng jīng, means “troll”), even though “干净” is typically a non-toxic word. False

¹<https://pypi.org/project/pypinyin/>

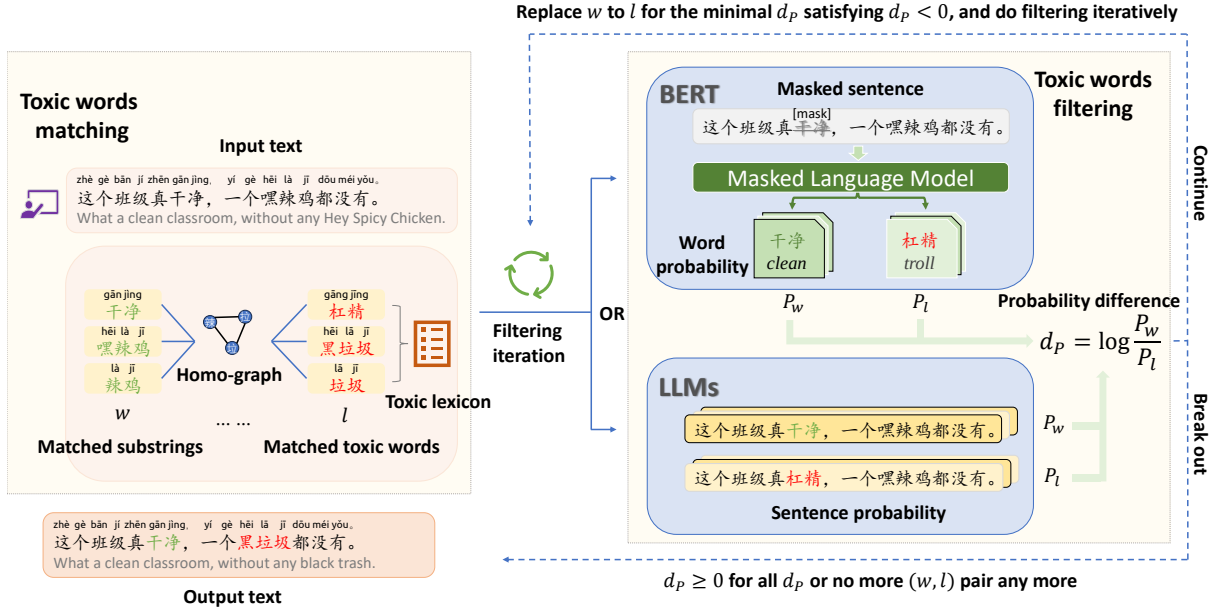


Figure 1: The main workflow of C²TU method. C²TU consists of two key stages: matching and filtering. In the matching stage, we identify candidate toxic terms in the input text by leveraging a homophone graph and a toxic lexicon. In the filtering stage, we use a language model (BERT model or LLM) to compute the probability gap for each (w, l) pair and iteratively replace the pair with the most significant gap. Once the iteration terminates, the unveiled sentence is returned.

positives are especially severe for single-character toxic words, as all homophones of the single toxic character (e.g., “操” (cào, means “f*ck”)) would be matched. The pseudocode of the matching algorithm is summarized in Alg. 1, which outputs M candidate toxic words pairs, denoted as $W_p = \{(w^{(i)}, l^{(i)})\}$ for $i = 1, \dots, M$. Here, $l^{(i)}$ is a probable toxic unveiling of $w^{(i)}$.

3.3 Filtering Candidate Toxic Words

To further filter out the incorrect matches (see Figure 2) in W_p , we leverage the full semantics of the text sequence X . Formally, for each (w, l) pair, we define X_{pre} and X_{tail} as the prefix sequence and suffix sequence of w in X , respectively. Given X_{pre} and X_{tail} , we next calculate the probabilities $P(w|X_{\text{pre}}, X_{\text{tail}})$ and $P(l|X_{\text{pre}}, X_{\text{tail}})$, shorted as P_w and P_l , respectively. For all pairs (w, l) that satisfy $P_w < P_l$, we select the one that yields the largest difference, denoted as $(\tilde{w}, \tilde{l})$. Formally, the probability difference is calculated by:

$$\text{ProbDiff}(P_w, P_l) = \log P_w - \log P_l = \log \frac{P_w}{P_l}. \quad (1)$$

After that, in the given text sequence X , we replace $\tilde{w}$ with $\tilde{l}$, remove $(\tilde{w}, \tilde{l})$ from W_p and then repeat the previous process until W_p is empty or all the remaining pairs in W_p have $P_w \geq P_l$. Note

that in each iteration, we need to recalculate the probabilities for each pair in W_p because once $\tilde{w}$ is replaced, X_{pre} and X_{tail} for remaining pairs change.

3.4 Computing Probability Difference

In this section, we introduce two methods for computing $P(x|X_{\text{pre}}, X_{\text{tail}})$, where x can be either w or l , with both BERT and LLMs.

3.4.1 BERT-based Method

During the pre-training phase, BERT utilizes an unsupervised Masked Language Modeling (MLM) task, whose goal is to predict tokens that are randomly masked. Hence, BERT is well-suited for computing $P(x|X_{\text{pre}}, X_{\text{tail}})$.

Given a text sequence $X = [X_{\text{pre}}, x, X_{\text{tail}}]$, for the i -th token x_i in x , we first mask it with the [mask] token and derive a text sequence $X_i^{\text{mask}} = [X_{\text{pre}}, \dots, x_{i-1}, [\text{mask}], x_{i+1}, \dots, X_{\text{tail}}]$. Then we can output the probability for the masked token $P_{\text{BERT}}(x_i|X_i^{\text{mask}})$. For a toxic word x that includes N tokens, we sequentially predict each of these tokens and combine them by:

$$P_{\text{BERT}}(x|X_{\text{pre}}, X_{\text{tail}}) = \prod_{i=1}^N P_{\text{BERT}}(x_i|X_i^{\text{mask}}). \quad (2)$$

Due to the varying word lengths, we adjust Equation 2 to calculate the geometric mean of all N

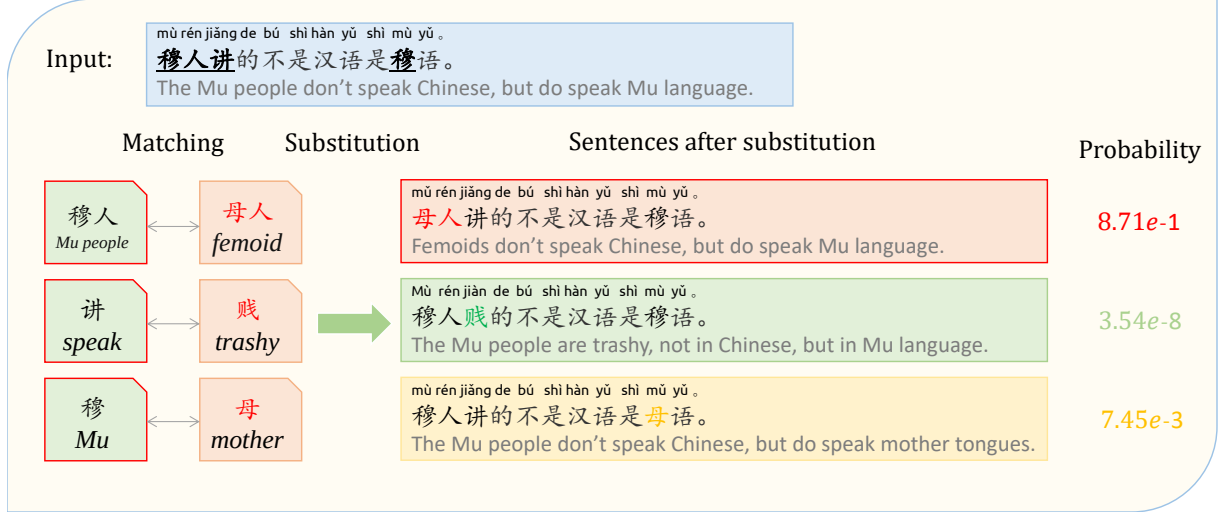


Figure 2: Three candidate toxic words pairs are matched in the input text. Each pair is individually substituted to generate three new sentences. It is evident that the pair [“讲”, “贱”] is incorrect, as the resulting sentence is highly unnatural in meaning, leading to a significantly smaller probability. In contrast, the substitutions of [“穆人”, “母人”] and [“穆”, “母”] are correct, and their corresponding sentences yield higher probabilities.

probabilities. Finally, the log probability to predict x is formally given by:

$$\log P_x = \log \sqrt[N]{P_{\text{BERT}}(x|X_{\text{pre}}, X_{\text{tail}})} \\ = \frac{1}{N} \sum_{i=1}^N \log P_{\text{BERT}}(x_i|X_i^{\text{mask}}) \quad (3)$$

3.4.2 LLMs-based Method

We next explore to use LLMs to filter candidate toxic words. Due to the auto-regressive nature, mainstream LLMs compute the occurrence probability of x based on the pre-context X_{pre} by:

$$P_{\text{LLM}}(x|X_{\text{pre}}) = \prod_{i=1}^N P_{\text{LLM}}(x_i|X_{\text{pre}}, x_1, \dots, x_{i-1}). \quad (4)$$

In this way, only pre-context is used while the post-context is totally ignored. To address the issue, instead of directly computing the probability for a word, we score each candidate by comparing the likelihood of the original sentence and the sentence after replacement. This yields a sentence-level form that can use the full context, as shown below. Formally, we show:

Theorem 1 Given a text sequence $X = [X_{\text{pre}}, w, X_{\text{tail}}]$ and a toxic word l with $\text{len}(w) = \text{len}(l) = N$, let X' denote the new sequence with w replaced by l . Then, the following relation holds:

$$\text{ProbDiff}(P_w, P_l) = \log P_{\text{LLM}}(X) - \log P_{\text{LLM}}(X'), \quad (5)$$

where $P_{\text{LLM}}(X)$ and $P_{\text{LLM}}(X')$ denote the probability of X and X' output by LLM, respectively. Both the probabilities can be calculated in the auto-regressive manner as Equation 4.

In this way, we can capture the semantics of the entire sentence, rather than only the pre-context information. The proof of Theorem 1 is provided in Appendix A.

3.5 Time Complexity Analysis

C²TU consists of two main stages: toxic word matching and filtering. Suppose we have an input text sequence X of length $|X|$ and a toxic lexicon with m toxic words.

In the matching stage, we enumerate substrings in X and match them with toxic words in the lexicon, whose largest length is N_{max} . We enumerate substrings of length $\leq N_{\text{max}}$ for matching. Considering that $N_{\text{max}} \ll |X|$, the number of enumerated substrings is linear to $|X|$. For each substring, we only need to match it with the toxic word of the same length, whose average number is assumed to be $\frac{m}{N_{\text{max}}}$. Therefore, the overall matching time complexity is $O(\frac{m}{N_{\text{max}}} |X|)$.

In the filtering stage, we iteratively handle M candidate (w, l) pairs. In each iteration, for each pair (w, l) , we calculate the occurrence probability difference of the two words, leading to a time complexity of $O(M|X|^2)$ no matter whether BERT or LLMs are used. The overall time complexity is thus $O(M^2|X|^2)$. Different from other LLM-based models that generate tokens with a time com-

plexity of $O(|X|^3)$, our method C²TU-LLM only uses LLMs to output the probability difference, which is highly more efficient.

4 Experiment

4.1 Experimental Details

Datasets We conduct experiments on ToxicloakCN (Xiao et al., 2024) and COLDataset (Deng et al., 2022). For ToxicloakCN, the main results reported in this paper use the homophone-cloaked subset. Emoji-based perturbations are excluded from the main tables and discussed as a limitation. Further, for fairness, we fine-tune the BERT-based CSC model on a subset of CHSD (Rao et al., 2023) named as CHSD-subset. Details on these datasets are given in Appendix C. We additionally provide an out-of-domain and cross-resource support analysis in Appendix D.

Baselines To our best knowledge, we are the first to unveil Chinese cloaked toxicity. Considering the relevance between our task and CSC task, we thus compare our method with three categories of baselines. (1) *Naive*: Directly replace all candidate toxic words without any filtering. (2) *BERT-based method*: SCOPE (Li et al., 2022) is a BERT-based CSC model that introduces an auxiliary task of Chinese pronunciation prediction to improve CSC task. We also finetune the pre-trained SCOPE model on CHSD-subset, obtaining the ft-SCOPE model. (3) *LLM-based method*: i) The prompt-based LLMs Baichuan2-7B-Base (Yang et al., 2023), Deepseek-V3 (Liu et al., 2024) and GPT-4o (Hurst et al., 2024). ii) Simple-CSC (Zhou et al., 2024) is a training-free and prompt-free LLM-based CSC model that considers character pronunciation and shape similarities.

Setup We construct homophone graph by pinyin, and utilize lexicons after manual correction and deduplication for toxicity matching. Our filtering model consists of two versions: BERT-based and LLM-based. For the former, we utilize bert-chinese-base (Devlin et al., 2019). For the latter, we use Baichuan2-7B-Base (Yang et al., 2023) as our backbone model. We provide a more detailed backbone comparison, including newer Qwen3.5 models, in Appendix H. None of the models’ parameters have been modified, nor have any prompts been used, as our method is entirely training-free and prompt-free. For SCOPE, we fol-

low its original pre-training stage and finetune it on CHSD-subset. For Simple-CSC, we employ it on Baichuan2-13B-Base as its original paper. All models can be downloaded on Huggingface². We also notice that there exist some other CSC baselines. However, some of them (Zhang et al., 2020; Dong et al., 2024) underperform the two used CSC baselines, while others (Cheng et al., 2020; Li et al., 2024) lack complete pretraining datasets and source codes. Thus, we exclude them from comparison. The prompt template used for prompt-based LLMs is presented in Appendix F. Our experiments are conducted on a server equipped with an Nvidia A800 80G GPU.

4.2 Main Results

We evaluate the model performance by F1 score and accuracy from two aspects (sentence-level and character-level), with each one containing both detection and correction tasks. All reported F1 and accuracy values are micro-averaged over the full evaluation set, following the aggregated counts defined in Appendix B. The sentence-level task requires the complete detection/correction of all Chinese cloaked characters in a sentence and keeps other non-cloaked characters not mis-detected/mis-corrected; otherwise, the task fails. For the character-level task, detection and correction are applied exclusively to individual characters. The effectiveness of the model is reflected by its ability to identify and correct more cloaked characters while minimizing modifications to non-toxic ones—the higher the true positive rate and the lower the false positive rate, the better the performance. Details on the evaluation metrics can be found in Appendix B. From Table 1, we observe:

(1) The Naive model generally performs poorly across the datasets, which shows the importance of candidate toxic words filtering.

(2) ft-SCOPE consistently outperforms SCOPE across all the metrics and datasets. This shows the significant domain gap between general CSC pretraining data and toxic speech corpora. C²TU-BERT, which is not fine-tuned, beats ft-SCOPE across all metrics. Although fine-tuning helps memorize toxic knowledge, it only memorizes fixed toxicity mapping patterns in the training data. However, our model injected with homophone graph and toxic lexicon has stronger generalizability, which can perceive more cloaked toxicities.

²<https://huggingface.co/>

Metric/%	Method	ToxicloakCN				COLDataset			
		Sentence		Character		Sentence		Character	
		Det.	Cor.	Det.	Cor.	Det.	Cor.	Det.	Cor.
F1 score	Naive	3.96	2.77	20.29	17.31	72.32	64.00	90.42	87.41
	SCOPE	13.23	6.33	7.81	2.22	14.80	9.91	18.41	7.01
	ft-SCOPE	35.79	26.99	17.48	11.73	62.66	59.15	41.21	35.83
	Simple-CSC	47.81	41.28	50.18	43.77	59.60	57.36	71.60	69.28
	Baichuan2-7B-Base	2.62	2.18	3.46	0.77	1.26	1.03	5.31	0.90
	Deepseek-V3	28.66	20.21	17.15	8.56	40.89	31.99	40.15	26.84
	GPT-4o	27.69	15.98	22.00	9.66	40.02	29.49	45.86	29.92
	C ² TU-BERT	<u>62.65</u>	<u>58.53</u>	<u>69.77</u>	<u>65.81</u>	<u>87.06</u>	<u>82.56</u>	<u>91.00</u>	<u>90.00</u>
	C ² TU-LLM	74.67	70.01	78.56	75.04	90.52	88.54	93.73	93.30
Accuracy	Naive	7.28	6.41	81.08	80.69	71.87	64.00	98.73	98.38
	SCOPE	35.35	31.33	79.42	78.79	15.47	11.22	78.35	76.78
	ft-SCOPE	52.88	47.36	80.72	80.01	63.52	60.14	82.32	81.33
	Simple-CSC	66.38	63.58	98.13	97.97	53.85	51.99	97.15	96.97
	Baichuan2-7B-Base	40.03	39.86	92.55	92.44	4.55	4.43	90.99	90.78
	Deepseek-V3	46.36	41.04	83.46	82.57	40.59	32.26	85.67	83.82
	GPT-4o	33.43	25.01	87.58	86.55	39.91	29.74	88.67	86.71
	C ² TU-BERT	<u>76.32</u>	<u>74.38</u>	<u>98.66</u>	<u>98.53</u>	<u>84.65</u>	<u>80.51</u>	<u>98.94</u>	<u>98.84</u>
	C ² TU-LLM	83.60	81.36	99.02	98.89	88.37	86.55	99.25	99.20

Table 1: The main results of all baseline models and our method w.r.t. the F1 score/% and the accuracy/% metrics, where Det. and Cor. mean detection and correction. We highlight the **best** score in bold and underline the runner-up one.

(3) The LLM-based model Simple-CSC exhibits undesired results in some cases, which are even worse than BERT-based models. For example, in the sentence-level toxicity detection on the COLDataset, Simple-CSC achieves an accuracy of 53.85% while that of ft-SCOPE is 63.52%. This can be attributed to the limited cloaked toxic knowledge of LLMs, which is in accordance with the poor performance of prompt-based LLM models Baichuan2-7B-Base, Deepseek-V3 and GPT-4o.

(4) C²TU-LLM and C²TU-BERT take the first two ranks in all contests. On the one hand, they use homophone graph and toxic lexicon to match all the potential toxic words in a given text sequence, which increases true positive rate. On the other hand, they further utilize a filtering step to mitigate the risk that transforms non-toxic words into toxic, which decreases false positive rate. We further corroborate this ordering with a human evaluation on 200 sampled ToxicloakCN examples, which is reported in Appendix I.

4.3 Analysis of Filtering Stage

As mentioned in Section 3.2, the Naive method that includes toxic word matching only and discards candidate toxic word filtering could lead to small false negative rate at the cost of high false

positive rate. We next show the results of *True/FALSE Positive/Negative rate, recall and precision* in Table 2. Naive has better TP and FN rates than our methods, which results in higher recall values because it regards all the candidate toxic substrings as the “true” toxic words. However, this leads to over-detection and over-correction, which categorizes a large number of non-toxic words to be toxic. Such indiscriminate filtering severely compromises the semantic integrity of benign sentences, making the Naive approach impractical for real-world scenarios where preserving the original meaning is crucial. Our methods further filter out candidate toxic words that are non-toxic and thus achieve better FP and TN rates, and higher precision scores. For example, On the ToxicloakCN dataset, in the correction task, Naive has a precision of 9.74%, while that of our model C²TU-LLM is 90.41%, which is around 10× larger. In summary, our methods significantly improve precision, at the cost of a mild drop in recall, which finally lead to larger F1 scores. Representative error cases for C²TU-BERT and C²TU-LLM are summarized in Appendix E.

4.4 Efficiency study

We next analyze the model efficiency. The results are presented in Figure 3, where ▲ indicates

Dataset	ToxicloakCN											
Metrics/%	Detection						Correction					
	TP↑	FP↓	FN↓	TN↑	recall↑	precision↑	TP↑	FP↓	FN↓	TN↑	recall↑	precision↑
Naive	2.41	18.74	0.19	78.67	92.74	11.39	2.02	18.74	0.58	78.67	77.83	9.74
C ² TU-BERT	1.55	0.29	1.05	97.11	59.54	84.25	1.42	0.29	1.18	97.11	54.50	83.04
C ² TU-LLM	1.79	0.18	0.80	97.23	69.09	91.04	1.67	0.18	0.93	97.23	64.13	90.41

Dataset	COLDataset											
Metrics/%	Detection						Correction					
	TP↑	FP↓	FN↓	TN↑	recall↑	precision↑	TP↑	FP↓	FN↓	TN↑	recall↑	precision↑
Naive	5.99	0.94	0.33	92.74	94.72	86.50	5.64	0.94	0.69	92.74	89.11	85.77
C ² TU-BERT	5.35	0.08	0.98	93.59	86.83	99.51	5.24	0.08	1.09	93.59	85.51	99.50
C ² TU-LLM	5.6	0.02	0.73	93.66	88.42	99.71	5.55	0.02	0.78	93.66	87.66	99.70

Table 2: Character-level analysis between the Naive method with C²TU on ToxicloakCN and COLDataset. T/F means True/False. P/N means Positive/Negative. All results are normalized by dividing the raw values by the total number of sentences and characters in the dataset. We highlight the **best** score in bold.

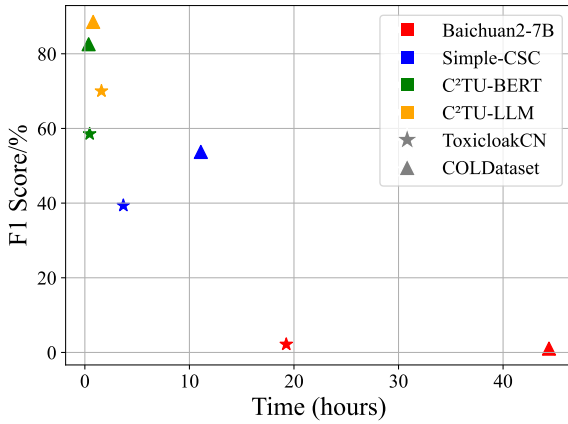


Figure 3: Efficiency study on the sentence-level correction task.

COLDataset and ★ represents ToxicloakCN. We use different colors to denote different methods including C²TU-LLM, C²TU-BERT, Simple-CSC and Baichuan2-7B-Base. For fairness, all LLM-based methods are based on Baichuan2-7B-Base. From the figure, we observe: (1) As LLM-based methods, C²TU-LLM is significantly more efficient than Simple-CSC and Baichuan2-7B-Base. This is because C²TU-LLM only uses LLM to output sentence probability difference, while the other two methods perform token generation. (2) C²TU-BERT is more efficient than C²TU-LLM due to the less number of model parameters, but it is at the cost of lower model effectiveness. In summary, our models are highly effective and also efficient.

4.5 Extension to Shape-Similar Cloaks

To further test whether C²TU is tied to homophone-based cloaks, we run a supplementary experiment on shape-similar characters. We keep the matching

Method	Sent-Det.	Sent-Cor.	Char-Det.	Char-Cor.
Simple-CSC	70.32	67.86	98.64	98.54
C ² TU-BERT	<u>89.97</u>	<u>89.97</u>	<u>99.42</u>	<u>99.42</u>
C ² TU-LLM	94.92	94.92	99.59	99.59

Table 3: Supplementary evaluation on ToxicloakCN with shape-similar cloaks. We extract the shape-similarity graph using SimilarCharacter and keep the rest of the framework unchanged.

and filtering pipeline unchanged and only replace the homophone graph with a shape-similarity graph extracted using the open-source SimilarCharacter³ library. This setting isolates whether the framework can transfer to another relation graph without additional training.

The results in Table 3 show that C²TU still brings consistent gains under a different relation graph, indicating that the framework is not limited to homophone cloaks alone.

4.6 Ablation Study

We also conduct an ablation study on C²TU to understand the characteristics of its main components. Given the auto-regressive nature of LLMs, it is direct to predict the occurrence probability of a word based only on its preceding context X_{pre} . This helps us understand the importance of the sentence probability method that leverages full context in the filtering stage. We call this method **C²TU-WP (Word Probability)**. Further, filtering candidate toxic words in Algorithm 2 involves multiple rounds. We introduce a single round strategy, in which each (w, l) pair satisfying $P_w < P_l$ is replaced. This helps us understand the importance of the iterative strategy for filtering. We call this

³<https://github.com/contr41/SimilarCharacter>

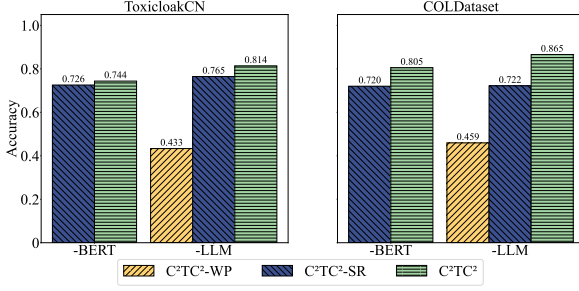


Figure 4: Sentence level correction accuracy gap between C^2TU , C^2TU -WP and C^2TU -SR on both datasets. The BERT-based model is employed on bert-chinese-base and the LLM-based model is employed on Baichuan2-7B-Base.

method C^2TU -SR (Single Round).

We then compare C^2TU with C^2TU -WP and C^2TU -SR. Due to the space limitation, we only show the results on the accuracy of sentence-level correction in Figure 4. For other cases, we observe similar results, detailed in Appendix J. From the figure, we see that: (1) C^2TU -LLM significantly beats C^2TU -WP. This indicates that X_{tail} also provides essential semantic information and leveraging the full text context for filtering candidate toxic words is necessary. (2) For both C^2TU -BERT and C^2TU -LLM, they consistently outperform C^2TU -SR across the datasets. This is because C^2TU replaces the (w, l) pair with the largest probability difference in each round, which iteratively enhances the context of remaining (w, l) pairs and the correction accuracy.

5 Conclusion

In this paper, we presented C^2TU , a training-free and prompt-free method for unveiling Chinese cloaked toxic contents on social platforms. Specifically, our method consists of two stages: Chinese cloaked toxicity matching and over-unveiling candidates filtering. For the former, we constructed the homophone graph to capture similar homophone relationships between Chinese characters. Subsequently, we matched toxic words via the homophone graph and toxic lexicon. To mitigate over-correction issues, we employed context-aware language models (BERT-based and LLM-based) to filter candidate toxic words iteratively. Comprehensive experimental results show that C^2TU achieves superior performance in Chinese cloaked toxicity unveiling task with high efficiency.

Limitations

Our method is designed to restore cloaked variants of known toxic terms, rather than discover entirely new toxic expressions or rapidly evolving slang. It mainly targets character-substitution cloaks, while other cloak types such as character-splitting, cross-lingual homophones, and emoji replacements remain outside the current framework and will be studied in future work.

Ethical Considerations

Our original intent is to unveil cloaked toxic contents online in order to optimize cyber governance and purify the digital environment, given that cloaked toxic contents are indeed a pervasive phenomenon.

We acknowledge that the toxic word candidates identified in Stage 1 - derived from homophonic graph and a toxic lexicon - while comprehensive, inevitably include numerous false positives. This occurs because simple homophonic matching with toxicity inherently produces substantial misclassifications, which will result in a significant number of normal expressions being misclassified as toxicity.

To address this, we implement our Stage 2 filtering method to systematically eliminate normal expressions that were mistakenly flagged. Our experimental results in Table 2 demonstrate that this filtering approach significantly reduces the false positive rate, albeit at some cost to true positive rate. This design choice prioritizes protecting users' rights to express legitimate content while maintaining detection efficacy.

From an academic research perspective, this represents the fullest extent of what we can achieve. Our technology is designed solely for academic research purposes, with no intended application in commercial or governmental contexts.

In the future, if this research ever be implemented in real-world content moderation systems, we maintain that human review remains indispensable. Our methodology can identify the vast majority of cloaked toxicity cases, which would then undergo human verification. This approach would ensure that every content flagged as toxic receives human validation.

To address the potential misuse of this technology for censorship, we require all users to obtain our authorization. We promise that we will not grant permission to government agencies or companies that are closely affiliated with governments.

Acknowledgments

This work was supported by Shanghai "Science and Technology Innovation Action Plan" Project under Grant No.23511100700.

References

- AI@Meta. 2024. [Llama 3 model card](#).
- Anjum and Rahul Katarya. 2024. Hate speech, toxicity detection in online social media: a recent survey of state of the art and opportunities. *International Journal of Information Security*, 23(1):577–608.
- Arnav Arora, Preslav Nakov, Momchil Hardalov, Sheikh Muhammad Sarwar, Vibha Nayak, Yoan Dinkov, Dimitrina Zlatkova, Kyle Dent, Ameya Bhatawdekar, Guillaume Bouchard, and 1 others. 2023. Detecting harmful content on online platforms: what platforms need vs. where research efforts go. *ACM Computing Surveys*, 56(3):1–17.
- Zewen Bai, Liang Yang, Shengdi Yin, Junyu Lu, Jingjie Zeng, Haohao Zhu, Yuanyuan Sun, and Hongfei Lin. 2025. State toxicon: A benchmark for span-level target-aware toxicity extraction in chinese hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10206–10219.
- Kiran Bala. 2014. Social media and changing communication patterns. *Global Media Journal: Indian Edition*, 5(1):1.
- Halyna Brykhanova, Veronika Zaitseva, Iryna Gamova, and Diana Fayvishenko. 2021. Social media as a new communication platform in the context of the information eco strategy. *Journal of Information Technology Management*, 13(Special Issue: Role of ICT in Advancing Business and Management):128–142.
- Ying Chen and Susan Guion-Anderson. 2011. Perceptual confusability of word-final nasals in southern min and mandarin: Implications for coda nasal mergers in chinese. In *ICPhS*, pages 464–467.
- Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. Spellgen: Incorporating phonological and visual similarities into language models for chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881.
- Chengyu Cui. 2020. [Jionlp](#).
- Thomas Davidson, Dana Warmusley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.
- Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. 2022. Cold: A benchmark for chinese offensive language detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11580–11599.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Bin Dong, Qingwei Zhao, and Yonghong Yan. 2006. Automatic scoring of flat tongue and raised tongue in computer-assisted mandarin learning. In *Proc. ISCSLP 2006*, pages 580–591.
- Guanting Dong, Jinxu Zhao, Tingfeng Hui, Daichi Guo, Wenlong Wang, Boqi Feng, Yueyan Qiu, Zhuoma Gongque, Keqing He, Zechen Wang, and 1 others. 2023. Revisit input perturbation problems for llms: A unified robustness evaluation framework for noisy slot filling task. In *CCF International conference on natural language processing and chinese computing*, pages 682–694. Springer.
- Ming Dong, Yujing Chen, Miao Zhang, Hao Sun, and Tingting He. 2024. Rich semantic knowledge enhanced large language models for few-shot chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7372–7383.
- Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *arXiv preprint arXiv:2404.05993*.
- Hongyu Gong, Yuchen Li, Suma Bhat, and Pramod Viswanath. 2019. Context-sensitive malicious spelling error correction. In *The World Wide Web Conference*, pages 2771–2777.
- Rhett Hanscom, Tamara Silbergleit Lehman, Qin Lv, and Shivakant Mishra. 2024. The toxicity phenomenon across social media. *arXiv preprint arXiv:2410.21589*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Nazanin Jafari, James Allan, and Sheikh Muhammad Sarwar. 2024. Target span detection for implicit harmful content. In *Proceedings of the 2024 acm sigir international conference on theory of information retrieval*, pages 117–122.
- Aiqi Jiang, Xiaohan Yang, Yang Liu, and Arkaitz Zubiaga. 2022. Swsr: A chinese dataset and lexicon for online sexism detection. *Online Social Networks and Media*, 27:100182.

- Jin Woo Kim, Andrew Guess, Brendan Nyhan, and Jason Reifler. 2021. The distorting prism of social media: How self-selection and exposure to incivility fuel online comment toxicity. *Journal of Communication*, 71(6):922–946.
- Hannah Kirk, Bertie Vidgen, Paul Röttger, Tristan Thrush, and Scott A Hale. 2022. Hatemoji: A test suite and adversarially-generated dataset for benchmarking and detecting emoji-based hate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1352–1368.
- Ettien Koffi. 2019. An acoustic phonetic account of the confusion between [l] and [n] by some chinese speakers.
- Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022. Improving chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 4275–4286.
- Kunting Li, Yong Hu, Liang He, Fandong Meng, and Jie Zhou. 2024. C-llm: Learn to check chinese spelling errors character by character. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5944–5957.
- Luiz Henrique Quevedo Lima, Adriana Silvina Pagano, and Ana Paula Couto da Silva. 2024. Toxic content detection in online social networks: A new dataset from brazilian reddit communities. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese-Vol. 1*, pages 472–482.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023. Facilitating fine-grained detection of chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 16235–16250.
- Ismail Adewale Omipidan and Bernice Oluwalanumi Sanusi. 2024. Rise of social media in the digital age: Whatsapp a threat to effective communication. *IMSU Journal of Communication Studies*, 8(1):142–153.
- Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.
- Xiaojun Rao, Yangsen Zhang, Qilong Jia, Xueyang Liu, Shuang Peng, and 1 others. 2023. Chinese hate speech detection method based on roberta-wwm. In *Proceedings of the 22nd Chinese national conference on computational linguistics*, pages 501–511.
- Raquel Recuero. 2024. The platformization of violence: Toward a concept of discursive toxicity on social media. *Social Media+ Society*, 10(1):20563051231224264.
- Maximilian Schmidhuber and Udo Kruschwitz. 2024. Llm-based synthetic datasets: Applications and limitations in toxicity detection. In *Proceedings of the Fourth Workshop on Threat, Aggression & Cyberbullying@ LREC-COLING-2024*, pages 37–51.
- Amit Sheth, Valerie L Shalin, and Ugur Kursuncu. 2022. Defining and detecting toxicity on social media: context and knowledge are key. *Neurocomputing*, 490:312–318.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#).
- Qwen Team. 2025. [Qwen3: Think deeper, act faster](#).
- Junqi Wu, Shujie Ji, Kang Zhong, Huiling Peng, Xiongding Liu, Wu Wei, and 1 others. 2025. Enhancing chinese offensive language detection with homophonic perturbation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22671–22686.
- Yunze Xiao, Yujia Hu, Kenny Tsu Wei Choo, and Roy Ka-Wei Lee. 2024. Toxicloackn: Evaluating robustness of offensive language detection in chinese with cloaking perturbations. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 6012–6025.
- Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, and 1 others. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.
- Shujian Yang, Shiyao Cui, Chuanrui Hu, Haicheng Wang, Tianwei Zhang, Minlie Huang, Jialiang Lu, and Han Qiu. 2025. Exploring multimodal challenges in toxic chinese detection: Taxonomy, benchmark, and findings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14382–14396.
- Hengxiang Zhang, Hongfu Gao, Qiang Hu, Guanhua Chen, Lili Yang, Bingyi Jing, Hongxin Wei, Bing Wang, Haifeng Bai, and Lei Yang. 2024. Chinesesafe: A chinese benchmark for evaluating safety in large language models. *arXiv preprint arXiv:2410.18491*.
- Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 882–890.

Yibo Zhao, Jiapeng Zhu, Can Xu, Yao Liu, and Xiang Li. 2025. Enhancing llm-based hatred and toxicity detection with meta-toxic knowledge graph. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24747–24760.

Houquan Zhou, Zhenghua Li, Bo Zhang, Chen Li, Shaopeng Lai, Ji Zhang, Fei Huang, and Min Zhang. 2024. A simple yet effective training-free prompt-free approach to chinese spelling correction based on large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17446–17467.

Jingyan Zhou, Jiawen Deng, Fei Mi, Yitong Li, Yasheng Wang, Minlie Huang, Xin Jiang, Qun Liu, and Helen Meng. 2022. Towards identifying social bias in dialog systems: Framework, dataset, and benchmark. In *Findings of the association for computational linguistics: EMNLP 2022*, pages 3576–3591.

Yuyan Zhou and Ying Wu. 2019. An erp study for phonemic merger in chinese dialects. *Modern Linguistics*, 7(1).

A Theoretical Proof

Proof A.1 Given a text sequence $X = [X_{pre}, w, X_{tail}]$ and a toxic word l with $\text{len}(w) = \text{len}(l) = N$, let X' denote the new sequence with w replaced by l . Then, the following equation holds:

$$\text{ProbDiff}(P_w, P_l) = \log P_{LLM}(X) - \log P_{LLM}(X') \quad (6)$$

where $P_{LLM}(X)$ and $P_{LLM}(X')$ denote the probability of X and X' output by LLM, respectively. Both the probabilities can be calculated in the autoregressive manner as Equation 4.

Firstly, we derive the general formula for $P_{LLM}(w|X_{pre}, X_{tail})$:

$$\begin{aligned} P_{LLM}(w|X_{pre}, X_{tail}) &= \frac{P_{LLM}(X_{pre}, w, X_{tail})}{P_{LLM}(X_{pre}, X_{tail})} \\ &= \frac{P_{LLM}(X)}{P_{LLM}(X_{pre}, X_{tail})} \end{aligned} \quad (7)$$

Analogously:

$$P_{LLM}(l|X_{pre}, X_{tail}) = \frac{P_{LLM}(X')}{P_{LLM}(X_{pre}, X_{tail})} \quad (8)$$

Therefore:

$$\begin{aligned} &\text{ProbDiff}(P_w, P_l) \\ &= \log P_w - \log P_l \\ &= \log P_{LLM}(w|X_{pre}, X_{tail}) \\ &\quad - \log P_{LLM}(l|X_{pre}, X_{tail}) \\ &= \log \frac{P_{LLM}(X)}{P_{LLM}(X_{pre}, X_{tail})} \\ &\quad - \log \frac{P_{LLM}(X')}{P_{LLM}(X_{pre}, X_{tail})} \\ &= \log \frac{P_{LLM}(X)}{P_{LLM}(X')} \\ &= \log P_{LLM}(X) - \log P_{LLM}(X') \end{aligned} \quad (9)$$

Q.E.D.

B Evaluation Metrics

We evaluate the model performance by F1 score and accuracy from two aspects (sentence-level and character-level), with each one containing both detection and correction tasks, which is common metrics in the CSC field. Unless otherwise stated, all reported F1 and accuracy values are micro-averaged over the full evaluation set, with TP, FP, FN, and TN accumulated before computing the metrics.

For sentence level, given the sample pair of (source sentence and target sentence) and the corresponding corrected sentence, if the length of corrected sentence is not equal to the target sentence, the result is FN for both detection and correction tasks. Otherwise, we have the following definition:

Sentence Level Detection (1) If the source sentence is different to the target sentence, the source sentence needs to be corrected. In this situation, if all wrong characters in source sentence is corrected (no matter correctly or not), and no proper character is over-corrected, the result is TP. Otherwise, the result is FN. (2) If the source sentence is identical to the target sentence, the source sentence doesn't need to be corrected. In this situation, if the corrected sentence is same to the target sentence, the result is TN. Otherwise, the result is FP.

Sentence Level Correction (1) If the source sentence is different to the target sentence, the source sentence needs to be corrected. In this situation, if all wrong characters in source sentence is corrected correctly, and no proper character is over-corrected, the result is TP. Otherwise, the result is FN. (2) If the source sentence is identical to the

target sentence, the source sentence doesn't need to be corrected. In this situation, if the corrected sentence is same to the target sentence, the result is TN. Otherwise, the result is FP.

For character level, given the sample pair of (source character and target character) and the corresponding corrected character:

Character Level Detection (1) If the source character is different to the target character, the source character needs to be corrected. In this situation, if the corrected character is not same to the source character (no matter the correction is correct or not), the result is TP. Otherwise, the result is FN. (2) If the source character is identical to the target character, the source character doesn't need to be corrected. In this situation, if the corrected character is same to the target character, the result is TN. Otherwise, the result is FP.

Character Level Correction (1) If the source character is different to the target character, the source character needs to be corrected. In this situation, if the source character is corrected correctly, the result is TP. Otherwise, the result is FN. (2) If the source character is identical to the target character, the source character doesn't need to be corrected. In this situation, if the corrected character is same to the target character, the result is TN. Otherwise, the result is FP.

Based on these definitions, we can calculate the F1 score (F_1) and the accuracy (Acc) by:

$$F_1 = \frac{2pr}{p+r} \quad (10)$$

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (11)$$

where $p = \frac{TP}{TP+FP}$ and $r = \frac{TP}{TP+FN}$.

C Datasets

We conduct experiments on two public datasets: ToxicloakCN (Xiao et al., 2024) and COLDataset (Deng et al., 2022), and finetune SCOPE (Li et al., 2022) model on CHSD-subset constructed from CHSD (Rao et al., 2023). Details on these datasets are given in Table 4.

Homo-Graph We use pypinyin to capture tone-ignored pinyin of all Chinese characters and connect characters with identical or phonetically similar pinyins, which is defined in Section 3.1.1. The details of homophone graphs are shown in Table 5.

Dataset	Size	Lexicon size
ToxicloakCN	4,586	387
COLDataset	10,415	115
CHSD-subset	2,790	602

Table 4: Dataset size and lexicon size of ToxicloakCN, COLDataset and CHSD-subset.

Dataset	Node size	Edge size
ToxicloakCN	3,613	141,683
COLDataset	3,879	163,969

Table 5: Homo-graph size of ToxicloakCN and COLDataset.

Toxic Lexicon (1) ToxicloakCN is an enhanced dataset derived from ToxiCN (Lu et al., 2023), where JioNLP (Cui, 2020) and NMSL⁴ are applied to perform homophone substitution and semantically similar emoji substitution on toxic speech, respectively (Xiao et al., 2024). For the experiments in this paper, we use the homophone-substitution subset only, and exclude emoji-based perturbations from the reported main results. For keyword-based substitution, the toxic lexicon from ToxiCN is used as keywords. For cleaner lexicon without cloak, we further correct and deduplicate the ToxiCN's lexicon. We first manually correct all toxic words in the original lexicon back to their proto-toxic words, and then retain each proto-toxic word. (2) COLDataset is collected through web crawling based on keywords, comprising 37,480 contents and spanning a wide range of topics related to race, gender, and regional issues. Since no existing work has compiled a toxic lexicon for COLDataset, we instead utilize the crawler keywords used during dataset construction (Deng et al., 2022) as toxic lexicon. Based on the lexicon, we merge the validation set with the test set and apply JioNLP (Cui, 2020) to perform homophone substitution and finally get the cloaked dataset, following the noise injection procedure of ToxicloakCN (Xiao et al., 2024).

	w/ T.L	w/o T.L
w/ C.L	1,706	542
w/o C.L	542	0

Table 6: The distribution of the CHSD-subset. T.L means ToxiCN's lexicon, while C.L means COLDataset's lexicon.

⁴<https://github.com/THUzhangga/NMSL>

Dataset for Finetuning To ensure fair comparison between our model and CSC model, we further fine-tune BERT-based CSC model SCOPE (Li et al., 2022) on a subset of CHSD (Rao et al., 2023), which integrates datasets COLDataset (Deng et al., 2022), SWSR (Jiang et al., 2022) and CDIAL-BIASDATASET (Zhou et al., 2022) to balance the distribution difference of toxic contents (see Table 6). To construct the CHSD-subset, we first filter out samples that appear in the part of COLDataset used in the test set. Next, we select samples based on toxic keywords from the two lexicons of ToxicloakCN and COLDataset, ensuring that our fine-tuning data contains relevant toxic content. Then we merge the two lexicons and apply JioNLP (Cui, 2020) to introduce noise. Note that we use the original lexicon of ToxiCN here. Finally, we construct CHSD-subset with 2790 samples that consists of 80% training and 20% validation data. Then we finetune the pre-trained SCOPE model with following settings: learning rate= $5e-5$, batch size=32, accumulate grad batches=2, epoch=20, warmup proportion=0.1. We first train the SCOPE model and it achieves a character-level detection F1-score of 0.8007 and a character-level corection F1-score of 0.7810 on CSC dataset. We then conducted fine-tuning on CHSD-subset and it achieves a character-level detection F1-score of 0.6026 and a character-level correction F1-score of 0.5531. Details of CHSD-subset are presented in Table 6.

D Out-of-Domain and Cross-Resource Analysis

To provide additional support for the robustness of our matching-and-filtering framework, we examine how much overlap exists between the homophone graphs built from COLDataset and ToxiCN, as summarized in Table 7. Although the two datasets differ, their commonly used character sets overlap substantially.

We further conduct a cross-resource experiment on ToxicloakCN using the ToxiCN lexicon, as shown in Table 8. We compare three homophone graphs: the original ToxiCN graph, a joint graph formed by taking the union of the COLDataset and ToxiCN graphs, and a general graph built from all CJK Unified Ideographs in U+4E00–U+9FFF using pypinyin. The general graph contains 20,992 nodes and 4,460,324 edges.

The results suggest that a broader graph does not necessarily improve performance, since the

Quantity	Value
COLDataset graph nodes	3,879
ToxiCN graph nodes	3,613
Intersection	3,118
Union	4,374
COLDataset-only	761
ToxiCN-only	495
Jaccard similarity	0.7128

Table 7: Overlap between the homophone graphs constructed from COLDataset and ToxiCN.

Method	Sent-Det.	Sent-Cor.	Char-Det.	Char-Cor.
w/ ToxiCN graph	74.67	70.01	78.56	75.04
w/ Joint graph	72.33	68.44	82.97	74.69
w/ General graph	72.33	68.44	82.97	74.69

Table 8: Cross-resource evaluation on ToxicloakCN using the ToxiCN lexicon. We report F1 scores (%).

contextual filter can already suppress many rare or implausible candidates. More broadly, Chinese characters and their pronunciations form a finite and relatively stable inventory, so temporal drift is more likely to affect the toxic lexicon than the homophone relation itself.

E Representative Error Cases

We manually inspect aligned original-cloaked sentence pairs and summarize representative errors from C²TU-BERT and C²TU-LLM in Table 9.

These examples suggest that the remaining failures are mostly caused by phonetic ambiguity, dialectal overgeneralization, and overlapping candidate spans.

F Prompt Template

Our prompt-based framework utilizes the following template structure (optimized version, generated by GPT-4o) to unveiling the cloaked toxicity:

Dataset	Error type	Original Sentence	Cloaked Sentence	C ² TU-BERT	C ² TU-LLM
ToxicloakCN	Short-span phonetic confusion	连说说也不准？黑吹好猖獗	连说说也不准？嗨吹好猖獗	连说说也不准？嗨吹好猖獗	连说说也不准？黑吹好猖獗
COLDataset	Polyphonic ambiguity	冷暴力是不好，但是对男对女都是一样的啊，很多人强调用冷暴力对女生这个点，不理解	冷暴力是不好，但是对男对女都是一样的啊，很多人强调用冷暴力对女省这个点，不理解	冷暴力是不好，但是对男对女都是一样的啊，很多人强调用冷暴力对女性这个点，不理解	冷暴力是不好，但是对男对女都是一样的啊，很多人强调用冷暴力对女性这个点，不理解
ToxicloakCN	Dialectal overgeneralization	你这个南蛮编的故事挺好的	你这个南蛮编的故事挺好的	你这个南蛮编的故事挺好的	你这个南蛮编的故事挺好的
ToxicloakCN	Overlapping candidates	凭什么你觉得是女拳师你就用娘们侮辱人？[捂嘴]我还觉得你是男拳师呢，哪个词伺候？[捂嘴]	凭什么你觉得是女权师你就用娘们侮辱人？[捂嘴]我还觉得你是南犬师呢，哪个词伺候？[捂嘴]	凭什么你觉得是女权师你就用娘们侮辱人？[捂嘴]我还觉得你是男权师呢，哪个词伺候？[捂嘴]	凭什么你觉得是女权师你就用娘们侮辱人？[捂嘴]我还觉得你是男权师呢，哪个词伺候？[捂嘴]

Table 9: Representative error cases of C²TU-BERT (bert-chinese-base) and C²TU-LLM (Baichuan2-7B-Base).

Prompt template for prompt-based method.	
- System:	
任务:	
针对中文敏感信息中带掩盖的错别字纠正。	
要求:	
1. 需要严格保证输入句子和输出句子长度一致;	
2. 只需要返回纠正后的句子，不要输出任何其他内容。	
- User: <Input sentence>	
- Assistant: <Output sentence>	

Table 10: Prompt template for prompt-based method.

G Pseudocode of Matching and Filtering

G.1 Toxic Words Matching Algorithm

Algorithm 1 Toxic Words Matching

Input: User comment X , lexicon $\mathcal{L}$, homophone graph $\mathcal{G}$

Output: Potentially cloaked toxic words W_p

```

1:  $W \leftarrow \text{SlidingWindow}(X)$ ,  $W_p \leftarrow \emptyset$ 
2: for each  $w^{(i)}$  in  $W$ , each  $l^{(j)}$  in  $\mathcal{L}$  do
3:   if  $\text{len}(w^{(i)}) = \text{len}(l^{(j)})$  then
4:      $N \leftarrow \text{len}(w^{(i)})$ ,  $\text{flag} \leftarrow \text{True}$ 
5:     for each  $k \in \{1, 2, \dots, N\}$  do
6:       if  $\mathcal{G}.\text{HasEdge}(w_k^{(i)}, l_k^{(j)}) \neq 1$  then
7:          $\text{flag} \leftarrow \text{False}$ 
8:       end if
9:     end for
10:    if  $\text{flag} = \text{True}$  then
11:       $W_p \leftarrow W_p \cup \{(w^{(i)}, l^{(j)})\}$ 
12:    end if
13:  end if
14: end for
15: return  $W_p$ 

```

G.2 Filtering Toxic Words Algorithm

Algorithm 2 Filtering Toxic Words

Input: $X, W_p, P(\cdot)$

Output: Corrected text X'

```

1: Initialize  $X' \leftarrow X$ 
2: while  $W_p$  is not empty do
3:   for each tuple  $(w^{(i)}, l^{(j)})$  in  $W_p$  do
4:      $X_{\text{pre}}, X_{\text{tail}} \leftarrow \text{GetPreTail}(X', w^{(i)})$ 
5:      $P_{w^{(i)}} \leftarrow P(w^{(i)} \mid X_{\text{pre}}, X_{\text{tail}})$ 
6:      $P_{l^{(j)}} \leftarrow P(l^{(j)} \mid X_{\text{pre}}, X_{\text{tail}})$ 
7:      $d_i \leftarrow \text{ProbDiff}(P_{w^{(i)}}, P_{l^{(j)}})$ 
8:   end for
9:   Get the index with the most significant difference:  $k \leftarrow \arg\min_i d_i$ 
10:  if  $P_{w^{(k)}} < P_{l^{(k)}}$  then
11:     $X_{\text{pre}}, X_{\text{tail}} \leftarrow \text{GetPreTail}(X', w^{(k)})$ 
12:    Replace:  $X' \leftarrow [X_{\text{pre}}, l^{(k)}, X_{\text{tail}}]$ 
13:     $W_p \leftarrow W_p \setminus \{(w^{(k)}, l^{(k)})\}$ 
14:  else
15:    break
16:  end if
17: end while
18: return  $X'$ 

```

H Performance Across LLMs

We evaluate the performance of C²TU-LLM across different LLMs, including LLaMA3-8B (AI@Meta, 2024), Qwen2.5-7B (Team, 2024) and Baichuan2-7B (Yang et al., 2023). We take Simple-CSC and Deepseek-V3 as our baselines, as they are also LLM-based. The results are shown in Table 11. While Simple-CSC employs Baichuan2-13B-Base and Deepseek-V3 is 671B, our method consistently outperforms them with different LLMs of smaller parameter scales (7B and 8B). This further shows the effectiveness of our method with the homophone graph and Chinese toxic lexicon,

Metric/%	Method	Model	ToxicloakCN				COLDataset			
			Sentence		Character		Sentence		Character	
			Det.	Cor.	Det.	Cor.	Det.	Cor.	Det.	Cor.
F1 score	Simple-CSC	Baichuan2-13B	47.81	41.28	50.18	43.77	59.60	57.36	71.60	69.28
	Prompt-based	Deepseek-V3	28.66	20.21	17.15	8.56	40.89	31.99	40.15	26.84
	C ² TU-LLM	LLaMA3-8B	68.74	64.30	74.44	70.98	89.55	86.25	92.74	91.97
		Qwen2.5-7B	67.20	63.52	73.46	70.19	89.37	86.30	92.41	91.73
		Baichuan2-7B	74.67	70.01	78.56	75.04	90.52	88.54	93.73	93.30
Accuracy	Simple-CSC	Baichuan2-13B	66.38	63.58	98.13	97.97	53.85	51.99	97.15	96.97
	Prompt-based	Deepseek-V3	46.36	41.04	83.46	82.57	40.59	32.26	85.67	83.82
	C ² TU-LLM	LLaMA3-8B	79.83	77.71	98.85	98.73	87.02	83.98	99.14	99.06
		Qwen2.5-7B	78.81	77.04	98.80	98.69	86.94	84.12	99.10	99.03
		Baichuan2-7B	83.60	81.36	99.02	98.89	88.37	86.55	99.25	99.20

Table 11: The results of C²TU-LLM with different LLMs w.r.t. F1 score and accuracy metrics, where Det. and Cor. mean detection and correction. We highlight the **best** score in bold.

Model	Sent-Det.	Sent-Cor.	Char-Det.	Char-Cor.
Baichuan2-7B	74.67	70.01	78.56	75.04
Qwen3.5-4B	69.31	64.83	81.39	71.53
Qwen3.5-7B	71.01	66.65	81.82	72.52

Table 12: Recent Qwen3.5 backbone comparison on ToxicloakCN. We report F1 scores (%).

Model	Correct	Incorrect	Accuracy (%)	95% CI (%)
Simple-CSC	154	46	77.00	[70.69, 82.29]
C ² TU-BERT	168	32	84.00	[78.29, 88.43]
C ² TU-LLM	178	22	89.00	[83.91, 92.62]

Table 13: Human evaluation on 200 sampled examples from ToxicloakCN.

which inject knowledge of cloaked toxicity into LLMs. We also report Qwen3.5 backbones (Team, 2025) on ToxicloakCN in Table 12.

The newer Qwen3.5 models still lag behind C²TU-LLM with Baichuan2-7B on this benchmark, which suggests that the task benefit comes more from the matching-and-filtering design than from a newer general-purpose backbone alone.

I Human Evaluation

Given that our evaluation metric for unveiling is based on rule matching rather than semantics, some sentences may not be corrected to their original form even though the unveiled outputs are valid. Therefore, we conducted a human evaluation based on sampling. Specifically, we randomly select 200 samples from the ToxicloakCN dataset and perform unveiling using three models: Simple-CSC, C²TU-BERT, and C²TU-LLM, which are the three strongest systems under automatic evaluation.

For evaluation, we recruit three evaluators (two female and one male) with prior knowledge of Chinese toxicity. All of them are highly interested in cloaked toxicity topic and volunteer to undergo the experiment. They are recruited as Research Assis-

tants (RAs) and remunerated in accordance with the wage standards prescribed by local laws.

All RAs receive detailed task instructions and are shown the toxic lexicon. During assessment, each evaluator is presented with five versions of each sentence: the raw sentence, cloaked sentence, and the three model-generated unveiled sentences. They are then asked to label each model’s output: marking it as "1" if they believed the sentence contains no remaining cloaking, and "0" otherwise. The final evaluation result for each sentence is determined by majority voting among the three evaluators’ labels. Table 13 reports the final human evaluation results, while Table 14 reports inter-annotator agreement statistics for the same sample. The larger sample size reduces variance, and the same ranking remains: C²TU-LLM performs best, followed by C²TU-BERT and then Simple-CSC.

Model	Fleiss’ Kappa	Observed Agreement $\bar{P}$	Chance Agreement P_e
C ² TU-LLM	0.87	0.97	0.79
C ² TU-BERT	0.88	0.97	0.72
Simple-CSC	0.94	0.98	0.64

Table 14: Inter-annotator agreement for human evaluation.

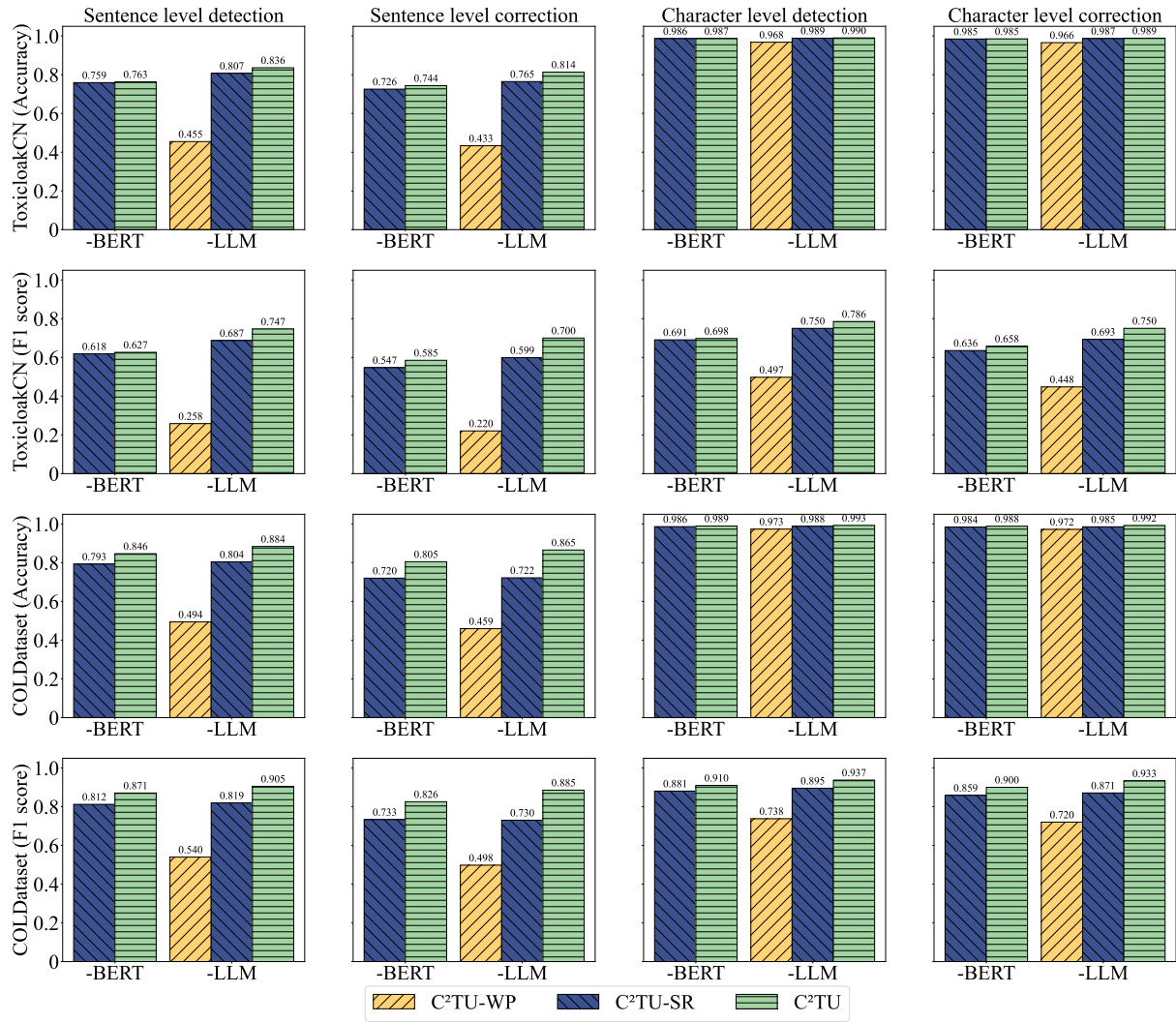


Figure 5: Full results of ablation study.

J Full Results of Ablation Study

Figure 5 summarizes the full ablation study between C²TU, C²TU-WP and C²TU-SR on both datasets. The C²TU-BERT is employed on bert-chinese-base and the C²TU-LLM is employed on Baichuan2-7B-Base.