Lightweight and Effective Coded-Slang Detection for Cyber-Drug Intelligence

Tao Leng , Yong Dai * , XinYang Yan

Posted Date: 13 May 2026

doi: 10.20944/preprints202605.0904.v1

Keywords: drug-related coded language; illicit content detection; TextCNN; deep learning; text classification

 doi:10.20944/preprints202605.0904.v1

Article

Lightweight and Effective Coded-Slang Detection for Cyber-Drug Intelligence

Tao Leng [1], Yong Dai [2,*] and XinYang Yan [1]

[1] Intelligent Policing Key Laboratory of Sichuan Province, Sichuan Policec College
[2] Department of Criminal Investigation, Sichuan Police College
* Correspondence: dy_scpc@163.com

Abstract

Drug-related criminal activities on social media increasingly employ dynamic coded language—such as fruit substitutions, numeric homophones, and dialectal metaphors—to evade detection. This linguistic obfuscation poses significant challenges to conventional keyword-based monitoring systems. Furthermore, the scarcity of open-source datasets capturing these specific evasive expressions severely impedes automated detection research. To address these limitations, we construct a dedicated dataset of 10000 samples of drug-related coded texts sourced from mainstream Chinese social media platforms. Concurrently, we propose an optimized, TextCNN-based deep learning framework tailored for the automated identification of such illicit content. By leveraging multi-scale convolutional feature extraction, our model effectively captures intricate local semantic patterns and morphological variations inherent in short, highly noisy social media texts. Experimental results demonstrate that the proposed method achieves an F1-score of 99.3%, significantly outperforming established baseline approaches in the semantic representation of coded language. These findings indicate that our framework provides an efficient, robust, and scalable computational solution for intelligent drug-related content monitoring in complex online environments.

Keywords: drug-related coded language; illicit content detection; TextCNN; deep learning; text classification

1. Introduction

The rapid proliferation of social media platforms has fundamentally transformed the operational landscape of drug-related criminal activities [1–5]. In China, the widespread misuse of synthetic drugs has been linked to a growing number of violent incidents, posing severe threats to public safety and social stability [6]. Unlike traditional offline drug transactions, illicit activities on platforms like Weibo, WeChat, and Xiaohongshu are deliberately concealed using coded language [7,8]. Common evasion strategies include euphemistic substitutions (e.g., fruits or animals), numeric homophones, and dialectal slang [9]. These strategies effectively circumvent platform-level keyword filtering and law enforcement surveillance. Consequently, a persistent gap is widening between criminal innovation and regulatory response.

This challenge is substantial on a global scale. The 2022 Annual Report of the European Monitoring Centre for Drugs and Drug Addiction (EMCDDA) highlights the rising prominence of social media in drug transactions [10]. Offenders are adopting more covert, intelligent, and diversified methods, significantly complicating investigative efforts. Similar trends exist in China, where drug-related terminology evolves rapidly within online communities. As a result, static rule-based detection systems quickly become ineffective against the continuous emergence of new slang expressions [11].

First, rule-based systems rely on manually curated lexicons and pattern matching [12]. While interpretable and easy to deploy, they suffer from poor generalization and high maintenance costs when facing constantly evolving coded language. Second, early machine learning methods, such as

 doi:10.20944/preprints202605.0904.v1

support vector machines (SVM) and Naïve Bayes, show limited capacity to capture contextual nuances [13]. This is especially problematic given the extreme brevity and high lexical noise of social media text. More recently, deep learning architectures—including RNN, LSTM, and BERT—have achieved strong performance on general text classification benchmarks [14–16].

However, applying these advanced models directly to Chinese drug-related coded language remains underexplored. This gap stems from two primary factors. First, there is a critical scarcity of domain-specific annotated datasets in Chinese. Second, social media drug slang possesses distinctive linguistic properties: extreme brevity, intentional semantic obfuscation, and rapid lexical turnover. Models designed for well-formed, resource-rich texts struggle to address these specific challenges [17,18].

Convolutional neural networks, particularly the TextCNN architecture proposed by Kim [19], offer a compelling solution for this low-resource setting. TextCNN applies parallel multi-scale convolutional filters over character- or word-level embeddings. This allows it to efficiently extract local n-gram semantic patterns from short, noisy texts. Crucially, it achieves this without requiring massive annotated datasets or extensive computational resources. Its architectural simplicity and strong empirical performance make it exceptionally well-suited for real-time drug slang detection on Chinese social media platforms [20].

To address the identified limitations and bridge the existing research gap, this study makes the following contributions:

1. **Domain-specific dataset construction.** We construct a dedicated, fully annotated dataset comprising 10000 drug-related coded language samples systematically collected from mainstream Chinese social media platforms. By making this resource open-source, we directly address the critical shortage of domain-specific Chinese corpora and provide a vital foundation for future illicit slang detection research.
2. **Low-resource detection framework.** We propose a TextCNN-based deep learning framework specifically tailored for Chinese drug-related coded language detection. By leveraging multi-scale convolutional filters at the character level, our model effectively captures local n-gram semantic features within short, highly noisy social media texts, proving exceptionally effective even under limited-annotation conditions.
3. **Systematic empirical evaluation.** We conduct comprehensive comparative experiments against multiple established baseline methods. The results demonstrate that our proposed framework achieves an outstanding F1-score of 99.3%, significantly outperforming conventional models. This validates its robustness, scalability, and practical effectiveness for deployment in real-world online drug surveillance systems.

The remainder of this paper is organized as follows. Section 2 reviews related work on text classification and drug-related content detection. Section 3 describes the dataset construction process. and presents the proposed TextCNN-based methodology. Section 4 reports experimental results and comparative analysis. Section 5 concludes the paper and outlines directions for future research.

2. Related Work

The automatic detection of drug-related slang involves three core dimensions: the linguistic cognition of slang, social media content mining, and low-resource text classification. While existing research has made progress across these areas, systematic solutions tailored to low-resource Chinese social media settings remain scarce.

2.1. Linguistic Patterns of Drug Slang

Understanding the linguistic patterns underlying drug-related slang is essential for building effective detection systems. Nahar et al. (2022) [3] reviewed drug-related slang from an interdisciplinary medico-linguistic perspective. They catalogued the types, origins, and evolution mechanisms of common drug terms, highlighting their highly covert nature and rapid rate of change. Sundaram

 doi:10.20944/preprints202605.0904.v1

et al. (2023) [6] conducted a systematic review of slang analytics on social media. They noted that slang is heavily context-dependent and spreads almost exclusively within closed communities rather than formal texts. This dynamic results in an acute scarcity of annotated corpora. Furthermore, Liu et al. (2019) [7] and Wu et al. (2018) [11] analyzed internet slang from a sentiment perspective. Liu et al. examined its persuasive effects in advertising, while Wu et al. constructed SlangSD, a sentiment lexicon for short texts. Collectively, these studies demonstrate that drug slang proliferates on Chinese social media via homophones, abbreviations, and coded language. Consequently, its low-resource nature and semantic ambiguity remain the primary obstacles to automated detection.

2.2. Word Embedding-Based Approaches

Word embedding methods represent the early dominant paradigm for slang detection. For instance, Holbrook et al. (2024) [8] applied Word2Vec to a Reddit corpus to construct semantic neighborhoods for drug slang. They identified candidate terms via vector similarity, achieving strong results in high-resource English settings. Gadusu and McGinty (2025) [12] systematically compared Word2Vec and BERT. They found that static word vectors struggle to disambiguate polysemous slang. Conversely, context-sensitive dynamic models offer clear advantages but demand extensive labeled data, causing performance to drop in low-resource scenarios. Shiozawa et al. (2026) [21] extended word co-occurrence statistics to dark web anchor texts. This approach partially mitigated the out-of-vocabulary problem. However, it relies on large-scale unlabelled corpora, which are extremely difficult to acquire from regulated Chinese social platforms. In summary, traditional word embedding methods face a fundamental data bottleneck in Chinese slang settings. Capturing local morphological features under limited annotation remains an open challenge.

2.3. Deep Learning in Slang Detection

The adoption of deep learning has substantially improved drug-related content detection. Hu et al. (2019) [17] proposed an ensemble framework to model drug abuse signals in sparse Twitter texts. This multi-model fusion enhanced robustness against noisy data, though it required higher data volumes. Tassone et al. (2020) [18] incorporated graph mining to model user relationships. This enriched the feature space, but graph-structural information is often inaccessible in anonymous online environments. Asim et al. (2025) [13] compared LSTM, CNN, and fine-tuned BERT on a Twitter dataset. They found that convolution-based models effectively extracted local n-gram features from short texts. Furthermore, CNNs converged more stably than large pre-trained models under limited sample sizes. Similarly, Hossain et al. (2018) [22] validated the effectiveness of local feature modeling for political slang discovery. Among these architectures, TextCNN (Kim, 2014) [19] has been widely validated for low-resource text classification. It applies parallel multi-scale convolutional filters to extract local semantic patterns and compresses them via global max-pooling. Building on these findings, this paper adopts TextCNN as the primary classification model. We adapt it specifically to character-level Chinese inputs and the unique morphological traits of drug slang, aiming for efficient detection in low-resource settings.

2.4. Large Language Models in Slang Detection

The rise of large language models (LLMs) offers new perspectives on slang understanding, but it also exposes clear limitations in low-resource scenarios. Sun et al. (2024) [23] evaluated mainstream LLMs on informal language tasks. They found that pre-trained models struggle with the contextual disambiguation of niche community slang. Carpenter et al. (2026) [24] validated LLM capabilities for English opioid slang. However, they noted that distributional biases in training corpora severely degrade performance on Chinese internet slang. Additionally, the SlangLLM framework proposed by Patel and Alsobeh (2025) [25] enables dynamic detection but demands substantial computational resources. This high overhead makes it unsuitable for real-time deployment. Compared to LLMs, lightweight discriminative models like TextCNN offer manageable training costs and lower inference latency. This makes them significantly better suited for practical drug slang monitoring in China.

 doi:10.20944/preprints202605.0904.v1

2.5. Scarcity of Chinese Datasets

Data scarcity is the fundamental bottleneck constraining progress in Chinese drug slang detection. For English tasks, the SlangTrack dataset (Aloraini et al., 2026) [26] provides an important benchmark for identifying slang usage. Unfortunately, no equivalent Chinese resource currently exists. Existing public drug datasets predominantly originate from English platforms like Twitter and Reddit. In contrast, labeled corpora from Chinese platforms (e.g., Weibo, WeChat) remain extremely scarce due to strict privacy protection and content regulation. This stark reality highlights the critical need for low-resource modeling methods. Consequently, it serves as the primary motivation for the dataset construction and model selection strategies adopted in this study.

3. Method

3.1. Overview

This study proposes a systematic pipeline for detecting drug-related coded language on social media platforms. As illustrated in Figure 1, the overall workflow consists of four primary stages:

Data Acquisition and Annotation: Raw text data are systematically collected from mainstream Chinese social media platforms (e.g., Weibo, Douyin, Bilibili, and Xiaohongshu) using automated web scrapers. Following rigorous noise reduction and data cleaning, the samples are manually annotated to distinguish illicit coded language from benign text, yielding a high-quality labeled dataset.

Feature Preprocessing: The annotated dataset is transformed into structured feature representations suitable for deep learning architectures. This stage encompasses tokenization, stop-word removal, and the generation of dense word embeddings.

Model Construction and Training: An optimized TextCNN-based classification framework is deployed. By applying multi-scale convolutional filters, the model efficiently extracts and aggregates local semantic features from the short, highly noisy texts characteristic of social media.

Performance Evaluation: The trained model is rigorously evaluated against established baseline methods using standard classification metrics to validate its robustness, scalability, and overall detection efficacy.

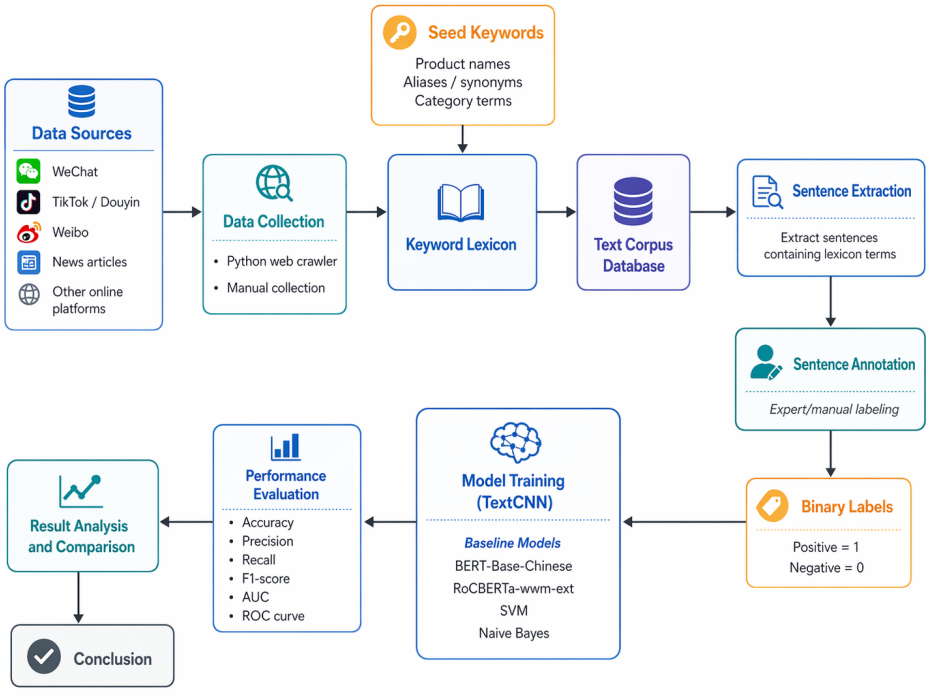

Figure 1. Architecture of Drug Slang Detection.

 doi:10.20944/preprints202605.0904.v1

3.2. Construction and Annotation of Drug-Related Slang Corpus

This chapter details the drug-related slang corpus construction process, including data sources, data preprocessing, data annotation, and statistical analysis, laying the foundation for subsequent model training and evaluation. The database construction flow chart is shown in Figure 2.

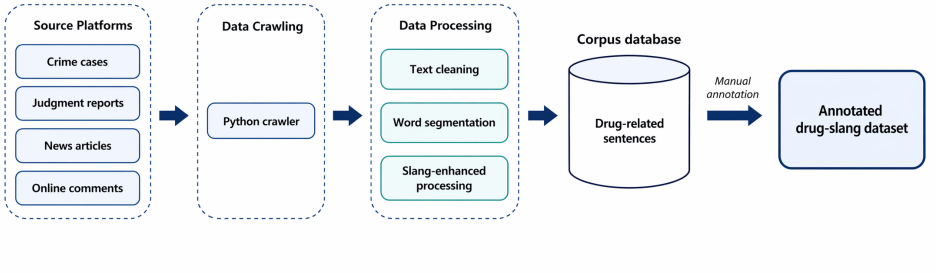

Figure 2. Flowchart of Database Construction.

3.2.1. Data Sources

This study employs a multi-source collaborative strategy for data collection, using two complementary approaches to obtain text data containing drug-related code language. First, Python web crawling techniques are used to collect publicly available interactive content from mainstream social platforms. The Sample data obtained by crawling are shown in Table 1.

Table 1. Examples of Data Collected via Web Crawling.

Sentences Containing Words from the Code Language Lexicon	**Source (Account)**
I'm a pilot, he's a captain, she's a farmer.	WeChat Official Account (People's Daily Online)
I'd like some dry pot — do you have any pork over there?	WeChat Official Account (Capital University Youth Red Ribbon)
I'm the captain. I've grown some grass. Any friends want to be pilots?	WeChat Official Account (Hubei Anti-Drug)
Electronic cigarettes, express delivery, high-altitude flight — no head rush, unconditional returns.	WeChat Official Account (Hubei Anti-Drug)
I'm the captain, I still have fuel here, looking for a few pilots. Do you have any "red wine" there? Newly arrived trousers with great cut, and top-grade red wine, welcome to try. Got some big meat — cold, hot, soft, hard, all kinds.	Weibo (China Anti-Drug Online) Weibo (Oriental Red People's Procuratorate) Douyin (Xinjiang Anti-Drug) Douyin (Yuelu Anti-Drug)
Want a late-night snack? Got some pork.	Douyin (Guangzhou Anti-Drug)

Data sources primarily include: parsing judicial documents from the Supreme People's Court Judgment Documents Network and drug-related case files from the Ministry of Public Security to extract drug-related code language patterns from real criminal cases; mining authoritative texts such as government work reports and anti-drug bulletins from public security agencies to capture policy-oriented drug-related semantic features; and dynamically crawling public drug-related posts from social platforms such as Weibo, Douyin, and WeChat using web crawlers, constructing a temporally varying corpus through sensitive information filtering and vectorization.

To augment the training data, this study automatically generated two categories of samples based on a drug-related code language lexicon: one category consists of sentences directly conveying hidden

 doi:10.20944/preprints202605.0904.v1

drug-related meanings (e.g., "A new batch of express tea has arrived"), and the other consists of contrastive sentences embedding code language vocabulary into positive contexts (e.g., "Community outreach on new tea varieties to prevent fraud"). By randomly combining different grammatical structures and contextual scenarios, tens of thousands of generated texts (over 5,000 drug-related sentences and over 5,000 normal sentences) maintain linguistic naturalness while covering expressive variation across diverse real-world scenarios. Partial examples of sentences randomly generated from the code language lexicon are shown in Table 2.

Table 2. Sample Sentences Randomly Generated from the Code Language Lexicon.

Sentences Containing Words from the Code Language Lexicon	**Code Language Words Included**
Have some "3-plus-1" tonight, let's have fun.	3-plus-1
My pigeons are great — dare to try?	Pigeons
This batch of "US dollars" is top quality, want some tonight?	US dollars
Want some G-water? Strong effects, give it a shot?	G-water
Bro, the "meow-meow" stock is plentiful lately — try some?	Meow-meow
This batch of "butterflies" is excellent — everyone try it.	Butterflies
Tonight, try this "Lamborghini" — guaranteed satisfaction.	Lamborghini
This batch of "Wuliangye" is absolutely top grade.	Wuliangye
Changzhijin is a harmful substance.	Changzhijin

3.2.2. Data Preprocessing

Raw text data collected from various sources often contains a large amount of noise and unstructured information that cannot be directly fed into a model for training. Therefore, a series of preprocessing operations must be applied to transform the raw data into standardized, clean text corpora. The preprocessing pipeline consists of three main steps: text cleaning, word segmentation, and code-language-specific augmentation.

(1)Text Cleaning: Text cleaning is the first step of preprocessing, aimed at removing noise information unrelated to semantics. Social media texts frequently contain URL links, HTML tags, emoticons, and various special symbols. This study uses regular expressions to filter out such content: for example, pattern matching is applied to remove URLs beginning with `http` or `www`, topic hashtag symbols are removed while preserving the topic text, and various punctuation marks are eliminated. Since some emoticons may be related to drug-related code language, they are converted to textual descriptions during cleaning to preserve potential semantic cues.

(2)Word Segmentation: Word segmentation is the process of dividing a continuous sequence of Chinese characters into word units with independent semantics, and can be understood as a mapping from a character sequence to a word sequence. Let the raw text be represented as a character sequence:

$$C = (c_1, c_2, \ldots, c_m) \tag{8}$$

where c_i denotes the i-th Chinese character. The goal of word segmentation is to convert this into a word sequence:

 doi:10.20944/preprints202605.0904.v1

$$W = (w_1, w_2, \ldots, w_n) \tag{9}$$

where w_j denotes a word composed of one or more consecutive characters. This process can be formally expressed as:

$$W = \text{Seg}(C, \mathcal{D}) \tag{10}$$

where $\text{Seg}(\cdot)$ denotes the segmentation function and $\mathcal{D}$ is the segmentation dictionary. This study uses the Jieba tokenizer to implement this mapping.

To address inaccurate segmentation of domain-specific vocabulary by general-purpose tokenizers, a domain-specific custom dictionary is introduced to extend the original dictionary:

$$\mathcal{D}' = \mathcal{D} \cup \mathcal{D}_{\text{drug}} \tag{11}$$

where $\mathcal{D}_{\text{drug}}$ contains the set of drug-related code language terms (e.g., "liubing," "zhurou," "linghao jiaonang," "xiaoqi," etc.). Under the extended dictionary constraint, the segmentation function is updated to:

$$W = \text{Seg}(C, \mathcal{D}') \tag{12}$$

By this means, words satisfying $w_j \in \mathcal{D}_{\text{drug}}$ are kept intact during segmentation, thus avoiding semantic fragmentation and improving recognition accuracy for domain-specific expressions. The custom dictionary is built upon the code language lexicon collected in the preliminary phase and is continuously expanded and refined throughout the annotation process.

(3)Augmentation Processing: To address the problem of variant expressions of drug-related code language on social media (such as homophone substitution, pinyin abbreviations, and numeric encoding), this study introduces semantic normalization on top of the segmentation results, mapping non-standard expressions to canonical forms [28]. A code language mapping function is constructed:

$$W' = \text{Norm}(W, \mathcal{M}) \tag{13}$$

where W' is the augmented word sequence. The mapping dictionary $\mathcal{M}$ transforms words one by one, defined as follows:

$$\mathcal{M}(w) = \begin{cases} m(w) & \text{if } w \in \text{dom}(\mathcal{M}) \\ w & \text{otherwise} \end{cases} \tag{14}$$

where $\text{dom}(\mathcal{M})$ denotes the domain of the mapping dictionary and $m(w)$ is the normalized expression corresponding to word w. For example, homophones or abbreviated forms are mapped to their standard code language equivalents. This process essentially implements a mapping from the original expression space to a canonical semantic space. Due to the polysemy and context-dependence of language, the mapping function is not strictly injective, and some words may carry semantic ambiguity. Nevertheless, in a statistical sense, this augmentation effectively improves the model's ability to recognize variant code language expressions, particularly yielding significant improvement in recall.

3.2.3. Data Annotation

Data annotation is the key step in converting raw text into usable samples. Annotation quality directly affects model learning outcomes. This study adopts a binary annotation scheme, classifying sentences as either non-drug-related (labeled 0) or drug-related (labeled 1). To ensure annotation accuracy and consistency, three annotators with relevant backgrounds were organized to participate in the labeling process. The annotation workflow is as follows:

 doi:10.20944/preprints202605.0904.v1

(1)Pre-annotation phase: A random sample of 200 instances is selected for pre-annotation. Inter-annotator agreement is calculated, disagreements are discussed, and consistent standards are established.

(2)Formal annotation phase: A double-blind annotation scheme is adopted, with each sample independently labeled by two annotators. Annotators do not communicate during the process to avoid mutual influence.

(3)Cross-review phase: Annotation results are compared; samples with consistent labels are directly accepted. For samples with inconsistent labels, a third senior annotator adjudicates to determine the final label.

To quantify the reliability of annotation results, this study uses Cohen's Kappa coefficient to assess inter-annotator agreement. The Kappa coefficient is calculated as:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{15}$$

where P_o is the observed agreement rate (the proportion of samples labeled identically by both annotators), and P_e is the expected agreement rate (the proportion of agreement attributable to chance). The Kappa value typically ranges from 0 to 1: values above 0.75 indicate good agreement, values between 0.40 and 0.75 indicate moderate agreement, and values below 0.40 indicate poor agreement. The Kappa coefficients at each annotation stage in this study are shown in Table 3.

Table 3. Cohen's Kappa Coefficients by Annotation Stage.

Annotation Stage	Kappa Coefficient
Pre-annotation	0.82
Formal annotation	0.91

The Kappa coefficients for the pre-annotation and formal annotation stages are 0.82 and 0.91, respectively, indicating extremely high inter-annotator agreement and confirming the reliability of the annotation results. Sample annotations from the dataset are shown in Table 4.

Table 4. Sample Sentence Annotations from the Dataset.

Sentence	Label
Want some new excitement? The effects are pretty good.	1
Coughed a couple of times this morning — mom and dad specially made cough syrup and brought it to school.	0
Heard this batch of cough syrup is premium quality, want to try some?	1
Bro, this batch of injections is way better than last time — let's arrange for tonight.	1
Went to the square today and saw the pigeons — they were so cute!	0
Man, this "little white" is very pure, want some?	1
Interested in tasting some "wind is tight"? I heard the flavor is pretty good.	1
How does the convenience-store coffee I just grabbed taste like cough syrup? Awful.	0
I've got fresh goods here, quality is beyond question — give it a try.	1

3.3. Model Architecture

3.3.1. Model Overview

proposes a domain-customized TextCNN framework tailored for drug-related code language detection. The overall model architecture is shown in Figure 3, comprising an embedding layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. The input consists of

 doi:10.20944/preprints202605.0904.v1

preprocessed short social media texts, and the output is a probability distribution indicating whether the text belongs to the drug-related or non-drug-related category.

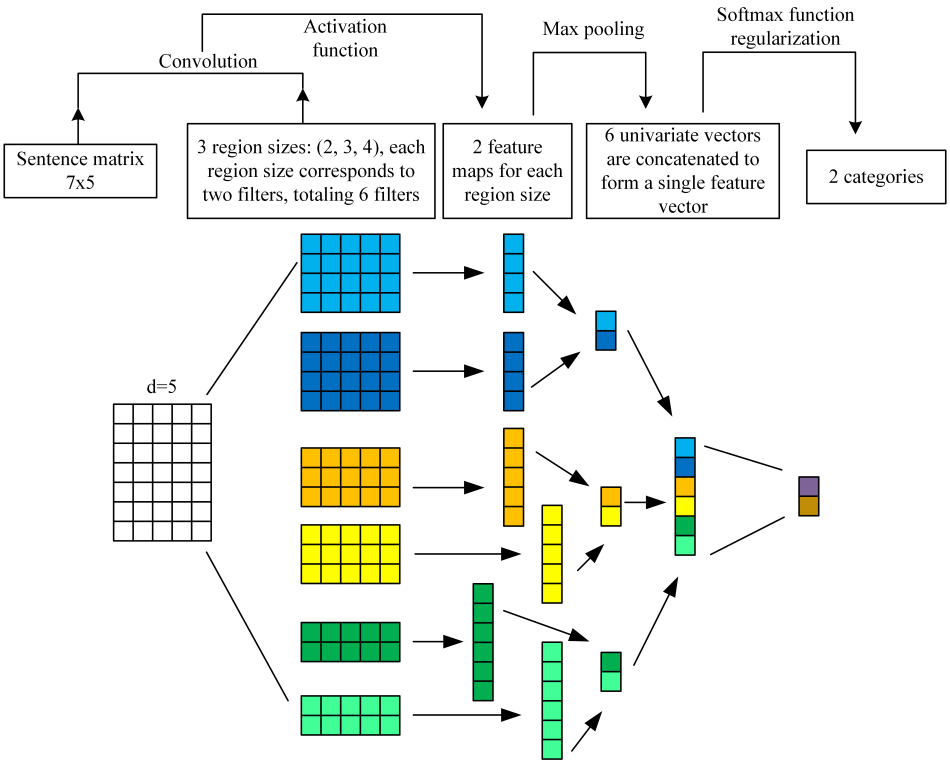

Figure 3. TextCNN Model Architecture.

The text sequence is first mapped through the embedding layer into 100-dimensional dense vectors, forming a word vector matrix. Subsequently, convolutional kernels of three sizes (3, 4, and 5) slide in parallel over the matrix to extract local semantic features of varying lengths, generating multiple sets of feature maps. A global max-pooling operation is then applied to each set of feature maps to retain the most salient feature responses, and all pooling results are concatenated into a fixed-length feature vector. Finally, this vector undergoes nonlinear transformation through the fully connected layer and is passed through a Softmax function to produce the classification output. During training, the Adam optimizer is used to update parameters, with the cross-entropy loss function employed to optimize the classification objective.

3.3.2. Input Layer and Embedding Layer

The input layer is responsible for receiving preprocessed sentence texts. Since neural networks require inputs of fixed dimensions, this study uniformly adjusts all text sequences to a length of 100 tokens. For sentences shorter than 100 tokens, zero-padding is applied at the end; for sentences longer than 100 tokens, only the first 100 tokens from the beginning are retained. This setting is based on statistical analysis of the dataset and is sufficient to cover the vast majority of short social media texts.

The embedding layer maps discrete word indices to continuous dense vectors. Suppose the input sentence contains n words ($n \leq 100$); the embedding layer converts it into a matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$, where d is the word vector dimensionality, and the i-th row of the matrix corresponds to the d-dimensional vector representation of the i-th word in the sentence. This study sets the word vector dimensionality to $d = 100$ and initializes the embedding matrix using pretrained Word2Vec word vectors [29]. Pretrained word vectors provide the model with rich semantic priors, enabling basic language understanding ability from the early stages of training. During training, the embedding layer parameters are set to a fine-tunable state, allowing the word vectors to adaptively adjust toward the semantic space of the drug-related domain.

 doi:10.20944/preprints202605.0904.v1

3.3.3. Convolutional Layer Design and Domain-Specific Adaptation

The convolutional layer is designed not merely for generic feature extraction, but as a specialized mechanism to penetrate the morphological camouflage of evasive illicit text. To effectively capture the dynamic and context-dependent patterns of coded expressions, this study employs parallel multi-scale convolutional filters. Specifically, the kernel heights are configured to $h \in \{3, 4, 5\}$.

The architectural decision to employ this specific configuration is deeply rooted in the adversarial nature of Chinese drug-related coded language. In online illicit markets, offenders deliberately utilize highly condensed, obfuscated terminology—ranging from two-character euphemisms (e.g., ice smoking", pork") to multi-character compound metaphors (e.g., Foxy Methoxy", Spice pen")—to circumvent static platform surveillance. During Chinese word segmentation, core two-character illicit phrases are frequently condensed into a single token. Consequently, setting the minimum kernel size to 3 serves as an optimal baseline: it captures not only the core illicit token itself but also its immediate syntactical anchors (e.g., surrounding verbs or prepositions). This tri-gram contextualization is critical for disambiguating benign usage from illicit intent in noisy environments.

Concurrently, larger convolutional filters ($h = 4$ and 5) are strategically deployed to capture longer, syntactically complex semantic obfuscations. These extended receptive fields are adept at recognizing fragmented expressions or multi-word idioms where criminals deliberately insert noise characters to bypass rule-based detection. By operating these three kernel sizes in parallel, the framework dynamically learns n-gram representations across varying granularities, establishing a robust defense against the continuous evolution of code-word length and structure.

Mechanically, each of the three kernel sizes comprises 128 independent filters. Given that the width of each filter strictly corresponds to the word vector dimensionality d, the weight matrix of an individual kernel is defined as $\mathbf{W} \in \mathbb{R}^{h \times d}$. During the forward pass, each kernel slides vertically across the input embedding matrix, processing h consecutive word vectors per stride. This operation computes a localized feature value via a dot product, followed immediately by a ReLU non-linear activation. Ultimately, each kernel generates a feature map whose sequence length is jointly determined by the input text length and the specific kernel height, effectively mapping raw adversarial text into a dense semantic feature space.

3.3.4. Pooling Layer and Feature Fusion

The pooling layer performs dimensionality reduction on the feature vectors output by the convolutional layer, extracting the most representative features. This study employs a global max-pooling strategy, selecting the maximum value from the feature vector generated by each convolutional kernel as that kernel's final output.

The core idea behind global max-pooling is that, for the drug-related code language recognition task, a key feature need only appear once in a sentence to support a judgment—its exact position is irrelevant. Regardless of whether "ice smoking" appears at the beginning, middle, or end of a sentence, it will be captured by the corresponding convolutional kernel, and the pooling layer retains its maximum activation value, thereby ensuring that critical information is not lost.

After the pooling layer, each convolutional kernel outputs a scalar value. With 128 kernels for each of the three sizes, a total of 384 scalar values are produced. These scalars are concatenated into a single feature vector that aggregates local semantic information at different granularities from the input sentence, providing a feature basis for subsequent classification.

3.3.5. Output Layer

The output layer consists of a fully connected layer and a Softmax classifier. The feature vector output by the pooling layer first undergoes nonlinear transformation and dimensionality compression through the fully connected layer. The fully connected layer has 128 neurons and uses the ReLU activation function:

$$\mathbf{h} = \mathrm{ReLU}(\mathbf{W}_{\mathrm{fc}}\,\mathbf{v} + \mathbf{b}) \tag{16}$$

 doi:10.20944/preprints202605.0904.v1

where $\mathbf{W}_{\text{fc}}$ is the weight matrix and $\mathbf{b}$ is the bias term.

To prevent overfitting, a Dropout mechanism is introduced after the fully connected layer. During training, Dropout randomly drops a fraction p of neuron outputs, forcing the model to avoid dependence on specific neurons. Dropout is active only during training; during inference, all neurons participate in computation.

The output layer applies the Softmax function to map the fully connected layer's output into binary classification probabilities:

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_{\text{out}}\,\mathbf{h} + \mathbf{b}_{\text{out}}) \tag{17}$$

The Softmax function is defined as $\text{Softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}$. The two output components correspond to the predicted probabilities for the non-drug-related (label 0) and drug-related (label 1) categories, respectively, and satisfy $\hat{y}_0 + \hat{y}_1 = 1$. The final prediction is the category with the higher probability.

3.3.6. Model Training Settings

After annotation, the dataset is randomly shuffled and split into training, validation, and test sets in an 8:1:1 ratio. The training set is used for learning and updating model parameters; the validation set monitors model performance during training; the test set is used for final evaluation of model performance.

Model training uses the Adam optimizer with an initial learning rate of 0.001. Adam combines the concepts of momentum and adaptive learning rates, dynamically adjusting the learning rate of each parameter based on estimates of the first and second moments of the gradients. It offers the advantages of fast convergence and low sensitivity to hyperparameters.

The loss function is cross-entropy loss, defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{p}_i \tag{18}$$

where N is the batch size, y_i is the true label of the i-th sample, and $\hat{p}_i$ is the model's predicted probability of the drug-related category. Cross-entropy loss effectively measures the discrepancy between the predicted probability distribution and the true label distribution.

The training batch size is set to 64, meaning 64 samples are processed simultaneously per iteration. The number of training epochs is set to 10. After each epoch, the loss and accuracy on the validation set are computed to monitor the training process. If validation performance ceases to improve, an early-stopping strategy is applied to prevent overfitting. The model with the best performance on the validation set is ultimately selected for evaluation on the test set. The main hyperparameters of this model are listed in Table 5.

Table 5. TextCNN Model Hyperparameters.

Parameter	Value
Maximum sequence length	100
Word vector dimensionality	100
Convolutional kernel sizes	3, 4, 5
Number of kernels per size	128
Pooling method	Global max-pooling
Fully connected layer neurons	128
Dropout rate	0.5
Optimizer	Adam
Initial learning rate	0.001
Loss function	Cross-entropy loss
Batch size	64
Number of training epochs	10
Train : Val : Test split	8:1:1

 doi:10.20944/preprints202605.0904.v1

4. Experiments

This chapter provides an experimental validation of the TextCNN model using a custom-built dataset of drug-related slang. We evaluate the model's performance through multi-dimensional metrics and conduct a comparative analysis against several baseline models. Furthermore, by performing an in-depth analysis of error cases and testing generalization capabilities, we offer a comprehensive examination of the model's performance and inherent limitations in practical application scenarios.

4.1. Experimental Setup

4.1.1. Implementation Environment

All experiments were conducted under a unified hardware and software configuration to ensure fair and reproducible comparisons across models. The detailed environment specifications are summarized in Table 6.

Table 6. Experimental environment configuration.

Configuration	Specification
Operating System	Ubuntu 20.04 LTS
CPU	Intel Xeon Gold 5218 @ 2.30 GHz
GPU	NVIDIA Tesla V100 32 GB
Memory	128 GB
Deep Learning Framework	PyTorch 1.10.0
Python Version	3.8.10
CUDA Version	11.3

4.1.2. Evaluation Metrics

Model performance was quantified using five standard evaluation metrics for binary text classification, computed from the four fundamental entries of the confusion matrix: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Accuracy measures the proportion of correctly classified samples over the entire test set:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

Precision measures the fraction of predicted positive samples that are truly positive, reflecting the reliability of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Recall measures the fraction of actual positive samples correctly identified by the model, reflecting coverage of the positive class:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score is the harmonic mean of precision and recall, providing a balanced measure of both correctness and completeness:

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

AUC (Area Under the ROC Curve) quantifies the model's ability to discriminate between positive and negative samples across all classification thresholds. An AUC value approaching 1.0 indicates near-perfect rank-ordering performance.

 doi:10.20944/preprints202605.0904.v1

4.1.3. Datasets Description and Partitioning

The experimental dataset employed in this study originates from the drug-related slang corpus constructed in the preceding section. Following rigorous data cleaning, tokenization, and manual annotation, the dataset provides a reliable foundation for model training and performance evaluation.

The dataset comprises a total of 10,000 text samples, with a balanced distribution between drug-related and non-drug-related instances. Data sources include legal documents, government bulletins, and various social media platforms—such as Weibo, TikTok (Douyin), and WeChat official accounts. To enhance the model's coverage across diverse linguistic styles, synthetic samples were generated by integrating a specialized slang dictionary. Consequently, the dataset captures not only direct drug-related jargon but also metaphorical expressions and linguistic variants, including homophones, abbreviations, and numeric substitutions, effectively reflecting the linguistic characteristics of real-world social media environments.

Regarding the labeling schema, a binary classification approach was adopted, where drug-related texts are labeled as 1 and non-drug-related texts as 0. This concise annotation system facilitates efficient discriminative learning for the model.

To ensure scientific rigor and reproducibility, the data were shuffled randomly and partitioned into training, validation, and test sets at a ratio of 8:1:1. The specific distribution is detailed in Table 7. To mitigate the impact of class imbalance on model training, the ratio of positive to negative samples remained consistent across all subsets, thereby ensuring the objectivity and stability of the evaluation results.

Table 7. Distribution of the experimental dataset.

Dataset	Sample Size	Proportion
Training Set	8,000	80%
Validation Set	1,000	10%
Test Set	1,000	10%

4.2. Performance Evaluation of TextCNN

4.2.1. Quantitative Results on Test Set

The trained TextCNN model was evaluated on the test set using a confusion matrix, from which accuracy, precision, recall, F1-score, and AUC were derived. Results are presented in Table 8.

Table 8. Performance metrics of the TextCNN model on the test set.

Metric	Value
Accuracy	0.9930
Precision	0.9930
Recall	0.9930
F1-Score	0.9930
AUC	0.9997

As illustrated in Table 8, the TextCNN model demonstrates exceptional performance in the drug-related slang recognition task. The model achieved an Accuracy of 99.30%, indicating that only approximately 7 errors occurred out of 1,000 test samples. Both Precision and Recall reached 0.9930, underscoring the model's high reliability in positive class prediction and its ability to capture the vast majority of actual drug-related instances. The F1-score, also recorded at 0.9930, signifies that the model maintains an optimal balance between precision and completeness. Furthermore, the AUC value of 0.9997—which significantly outperforms a random classifier—validates the model's near-perfect discriminative capability in distinguishing between positive and negative samples. The confusion matrix for the model on the test set is presented in Figure 4.

 doi:10.20944/preprints202605.0904.v1

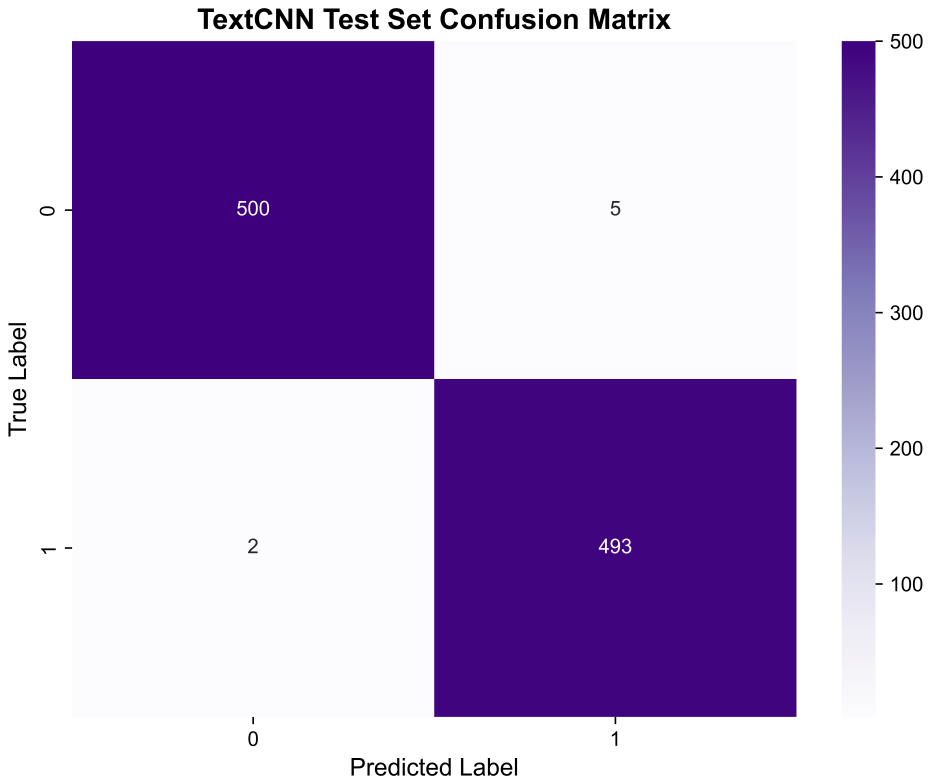

Figure 4. Confusion Matrix of the TextCNN Model on the Test Set.

4.2.2. Analysis of the Confusion Matrix

As demonstrated in the confusion matrix (Figure 4), the model produced only seven errors out of the 1,000 test samples. Specifically, these consisted of five false positives (misreporting) and two false negatives (omissions). These findings are highly consistent with the performance metrics presented in Table 7.

The confusion matrix provides visual verification of the model's exceptional discriminative capability for both drug-related slang and non-drug-related text. Furthermore, the error distribution is relatively uniform, indicating that the model does not exhibit a significant bias toward any specific category.

4.3. Comparative Analysis with Baseline Models

4.3.1. Performance Comparison on Classification Metrics

To verify the advantages of the TextCNN model in the task of identifying drug-related slang, this study selected four models—Support Vector Machine (SVM), Naive Bayes, BERT-Base-Chinese, and Chinese-RoBERTa-wwm-ext—for comparative experiments. All models were trained and evaluated on the same training set and test set, and the results are shown in Table 10.

Table 9. Performance comparison of different models on the test set.

Model	Accuracy	Precision	Recall	F1 Score	AUC
TextCNN	0.9930	0.9930	0.9930	0.9930	0.9997
BERT-Base-Chinese	0.9890	0.9892	0.9890	0.9890	0.9998
Chinese-RoBERTa-wwm-ext	0.9890	0.9892	0.9890	0.9890	0.9998
SVM	0.9690	1.0000	0.9374	0.9677	0.9907
Naive Bayes	0.9680	0.9978	0.9374	0.9667	0.9920

 doi:10.20944/preprints202605.0904.v1

To visually present the differences in key metrics among the models, Figure 5 shows a bar chart comparing the five models in terms of accuracy, recall, and F1 score. It can be clearly seen from the figure that the TextCNN model performs the best, achieving the highest values in all three metrics—accuracy, recall, and F1 score. On the recall metric, which is the most critical for the task of identifying drug-related slang, TextCNN outperforms traditional machine learning models such as SVM and Naive Bayes by approximately 5.6 percentage points, and also holds a slight lead of 0.4 percentage points over the pre-trained models BERT-Base-Chinese and Chinese-RoBERTa-wwm-ext. It is worth noting that although the SVM and Naive Bayes models achieve high precision (1.0000 and 0.9978, respectively), their recall is significantly lower, indicating a serious risk of missed detections.

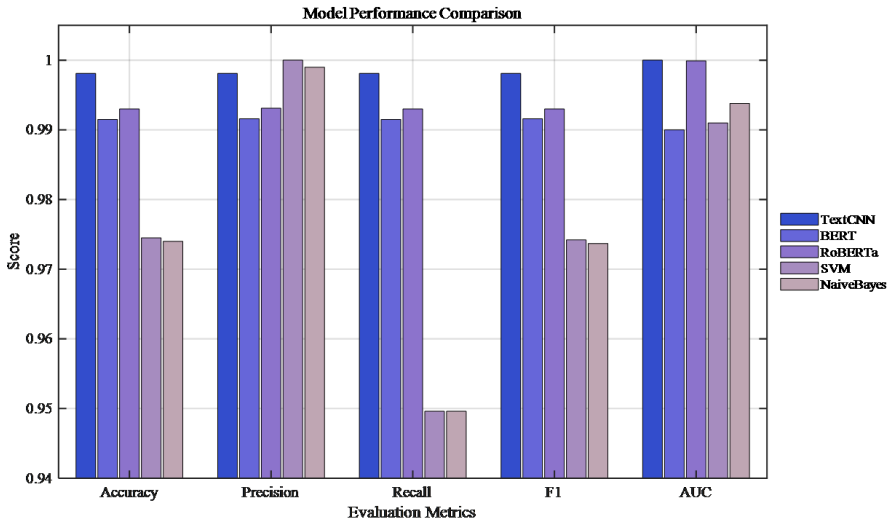

Figure 5. Performance comparison of different models.

To more comprehensively compare the overall performance of each model, Figure 6 presents a multi-indicator radar chart for the five models. It can be clearly seen from the figure that the polygon of the TextCNN model covers the largest area, leading in the three core indicators—accuracy, recall, and F1 score—thus demonstrating the best overall performance.

In contrast, although the SVM and Naïve Bayes models perform well in terms of precision, they show significant shortcomings in recall, with an apparent indentation in the polygon along this dimension. This reflects a serious issue of missed detections, rendering them unable to meet the practical requirement of "better false alarms than missed detections" in drug-related slang identification. The BERT-Base-Chinese and Chinese-RoBERTa-wwm-ext models exhibit relatively balanced overall performance, yet their coverage areas are slightly smaller than that of TextCNN, confirming the latter's comprehensive advantage in the task of identifying drug-related slang.

 doi:10.20944/preprints202605.0904.v1

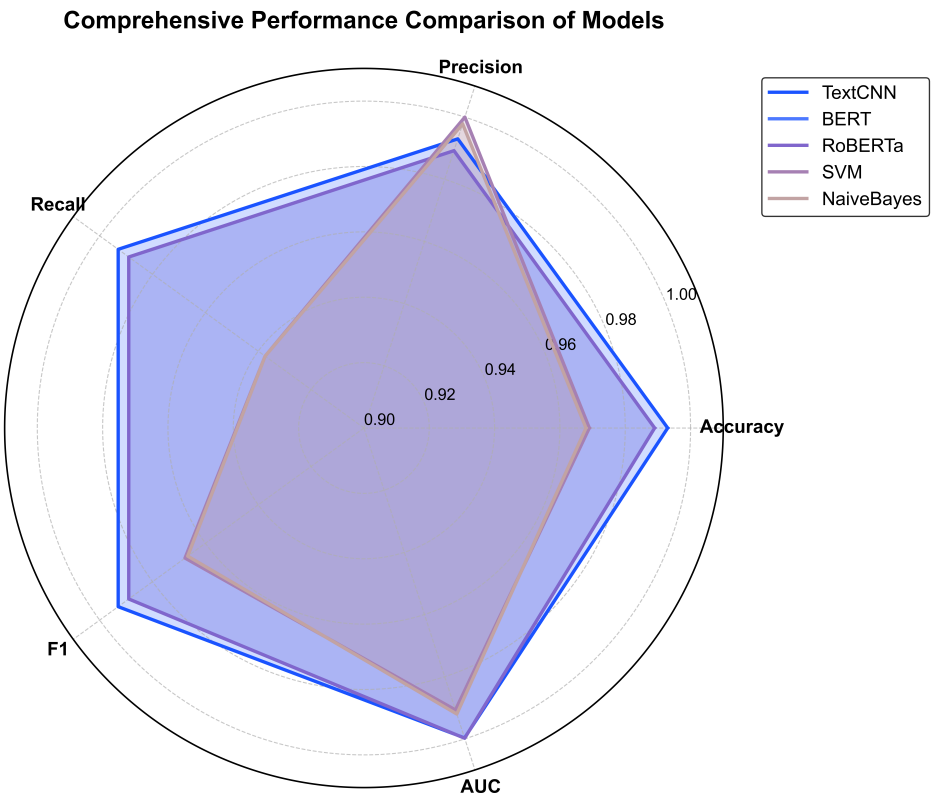

Figure 6. Radar chart of comprehensive performance.

4.3.2. Analysis of False Negatives (Missed Samples)

To further quantify the practical application value of each model, Figure 7 compares the number of missed detections of each model on the test set. It can be intuitively observed from the figure that the TextCNN model has only 7 missed detections, which is significantly lower than the 11 missed detections of BERT-Base-Chinese and Chinese-RoBERTa-wwm-ext, and merely about one-quarter of those of SVM and Naïve Bayes. In the task of detecting drug-related coded language, a single missed detection of drug-related information may lead to serious security consequences; therefore, the number of missed detections serves as a core indicator for measuring the practicality of a model. While maintaining leading classification performance, TextCNN achieves the lowest miss rate, fully demonstrating its deployment value in real public security scenarios. In contrast, although SVM and Naïve Bayes perform well in terms of precision, their excessively high number of missed detections renders them unable to meet the practical requirements of the task.

 doi:10.20944/preprints202605.0904.v1

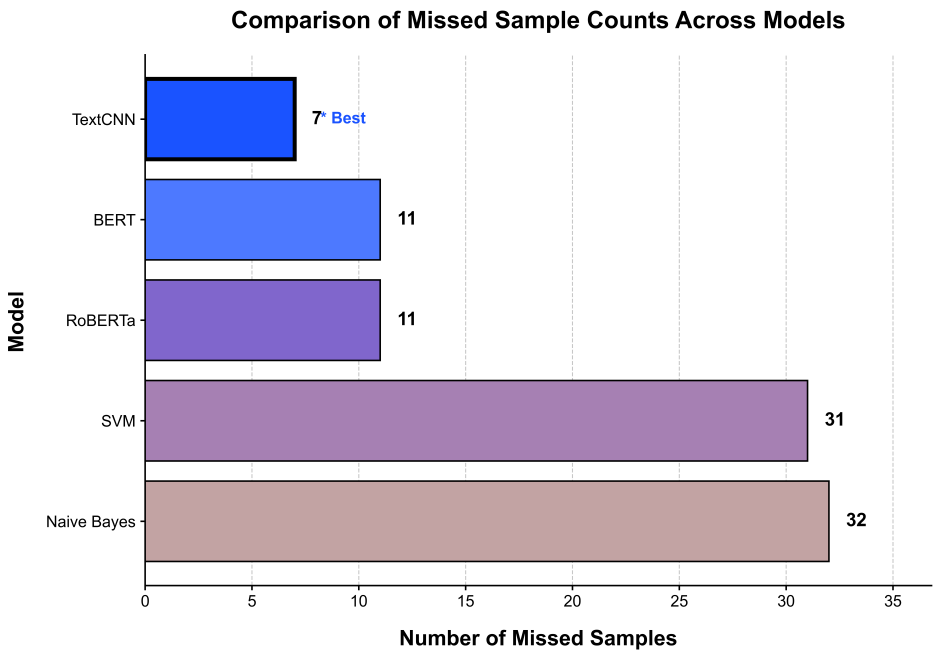

Figure 7. Comparison of missed sample counts across models.

4.3.3. Computational Efficiency and Deployment Costs

Figure 8 provides a comparative analysis of model sizes and inference speeds. Traditional machine learning models exhibit a distinct advantage in computational efficiency; for instance, the Naive Bayes model has a footprint of only 0.57 MB with a single-sample inference time of approximately 1.04 ms, while the SVM model measures 0.61 MB with an inference time of 0.90 ms. Although these traditional models show slightly lower classification performance than TextCNN, their minimal deployment costs and rapid inference speeds make them suitable for edge devices with limited computational resources and strict real-time requirements.

In contrast, while pre-trained models such as BERT achieve classification performance comparable to TextCNN, they suffer from significant disadvantages in deployment efficiency. The BERT-Base-Chinese model size is approximately 390.20 MB, with a single-sample inference time of 1269.25 ms (1.27 seconds) and a total time of 133.79 seconds for 1,000 samples—values that are 464 times, 682 times, and 73 times those of TextCNN, respectively. Such substantial model volume and slow inference speeds render BERT-Base-Chinese incapable of meeting the real-time detection demands of drug-related slang recognition. While Chinese-RoBERTa-wwm-ext offers improved speed (24.33 ms per sample), its size remains high at 390.20 MB, maintaining a high barrier for deployment.

TextCNN achieves an optimal balance between performance and efficiency. With a model size of approximately 0.84 MB, a single-sample inference time of 1.86 ms, and a 1,000-sample processing time of 1.82 seconds, it significantly outperforms pre-trained models in efficiency while maintaining substantially higher classification accuracy than traditional machine learning methods. These characteristics enable TextCNN to be deployed in real-time across various public security hardware environments, making it the most suitable candidate for practical law enforcement applications.

As indicated by the aforementioned results, in short-text classification tasks such as slang detection, TextCNN demonstrates superior recognition performance alongside significant advantages in inference efficiency and deployment costs compared to BERT-Base-Chinese, Chinese-RoBERTa-wwm-ext, SVM, and Naive Bayes. These discrepancies primarily arise from differences in model architectural characteristics, data dependency, and task alignment.

 doi:10.20944/preprints202605.0904.v1

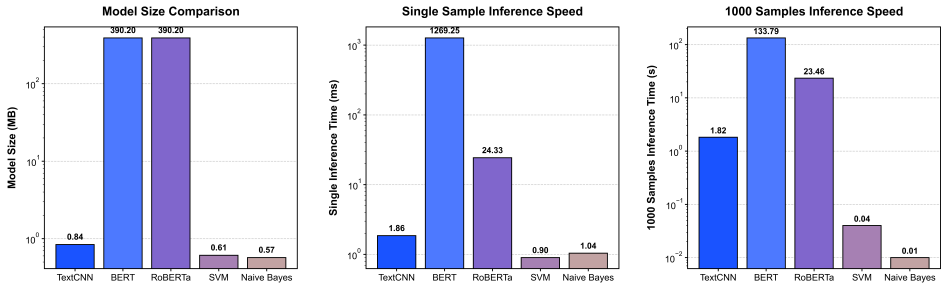

Figure 8. Model efficiency comparison.

4.3.4. Discussion on Model Structural Characteristics

The superior performance of TextCNN in terms of identification accuracy, inference efficiency, and deployment costs—relative to BERT-Base-Chinese, Chinese-RoBERTa-wwm-ext, SVM, and Naive Bayes—stems from inherent differences in model architecture, data dependency, and task alignment.

Structural Architectural Analysis: TextCNN is based on a Convolutional Neural Network (CNN) architecture that utilizes multi-scale convolutional kernels to extract local n-gram features. This mechanism effectively captures representative keyword combinations and local patterns, which aligns perfectly with the linguistic characteristics of drug slang, such as word substitutions and fixed collocations.

Conversely, Transformer-based pre-trained models like BERT and RoBERTa [30] rely on multi-layer self-attention mechanisms to model long-range semantic relationships and global contextual dependencies. While these models excel in complex tasks like semantic reasoning and reading comprehension, such sophisticated modeling is not always critical for slang detection. In this specific task, complex semantic modeling may introduce redundant information or semantic noise, potentially weakening the model's discriminative precision.

Data Dependency and Training Efficiency: Large-scale models like BERT and RoBERTa possess a massive number of parameters, requiring substantial annotated datasets and high-performance computational resources. When training samples are limited or when there is a significant domain discrepancy, these models are prone to overfitting or insufficient transfer learning. In contrast, TextCNN features a streamlined structure with fewer parameters. It achieves rapid convergence and efficiently learns task-specific features even under limited sample conditions, demonstrating superior training stability and data utilization.

Comparison with Traditional Models: Traditional machine learning models, such as SVM and Naive Bayes, primarily rely on manually engineered features like term frequency or TF-IDF. These methods lack the capacity to characterize nuanced contextual semantic relationships, making it difficult for them to handle the metaphorical expressions and semantic deformations prevalent in drug-related jargon.

In conclusion, TextCNN successfully balances feature extraction capability with a lightweight structural design. It achieves an optimal equilibrium between task adaptability, model complexity, and generalization performance. Consequently, in slang detection tasks dominated by local pattern recognition, TextCNN outperforms both large-scale pre-trained models and traditional shallow models. These disparities in efficiency and performance are the direct result of the specific structural characteristics and design objectives of each model.

4.4. Error Dissection and Case Study

To investigate the recognition boundaries of the proposed model and identify directions for future improvement, a qualitative analysis of misclassified samples from the test set was conducted. Despite the model's strong overall performance, systematic examination of prediction errors reveals potential limitations in semantic understanding. Representative error cases are summarized in Table 9, categorized into four distinct error types.

 doi:10.20944/preprints202605.0904.v1

Table 10. Error case analysis of the TextCNN model.

Original Text	True	Pred	Error Analysis	Error Type
I have some *xiǎo hǎi* here, want to try it?	1	0	Failure to recognize "*xiǎo hǎi*" as an explicit drug-related coded term	Coded language recognition failure
This batch of tin foil heats up really well, let's all try it together.	0	1	Tin foil is used in both normal cooking and drug consumption contexts; the model oversensitively classifies a culinary reference as drug-related	Context-induced misclassification
This *green* recipe is really simple and tastes great.	0	1	"green" may refer to vegetables (normal) or cannabis (drug); without sufficient context, the model favors the drug-related interpretation	Context-induced misclassification
Want to try the newly arrived *school uniform*? Works great.	0	1	"School uniform" (*jiào fú*) may serve as drug slang in specific subcultures but carries no such meaning here; the model overgeneralizes	Cultural expression misunderstanding
How many *bones* do you want? I have stock.	0	1	The model lacks the capacity to recognize humor and irony, defaulting to a literal interpretation of the utterance	Figurative language misinterpretation
I have some *cold stuff* here, very pure. Coming or not?	0	1	Likely a hyperbolic description of a normal item, but the model interprets it as drug-related coded language	Hyperbolic expression misinterpretation

The first error type is coded language recognition failure, characterized by false negatives in which the model fails to identify legitimate drug-related coded expressions. This typically occurs when certain slang terms appear infrequently in the training data or exhibit regional and temporal variation. For instance, the term "xiǎo hǎi", a regionally specific drug alias, was not successfully identified, suggesting that low-frequency or geographically localized coded expressions remain a coverage challenge for the current model.

The second error type is context-induced misclassification, where the model incorrectly interprets ambiguous terms due to insufficient contextual reasoning. For example, "tin foil" appears in both legitimate culinary contexts and drug consumption scenarios; without broader contextual understanding, the model relies solely on local lexical features, leading to false positives. Similarly, "green" may refer to vegetables or cannabis depending on context, and the model tends to favor the drug-related interpretation in the absence of disambiguating cues.

The third error type is cultural and community-specific expression misunderstanding. Certain terms carry drug-related connotations only within specific subcultures or demographic groups. The term "school uniform", for instance, may serve as coded drug language within particular communities but carries no such meaning in general usage. The model struggles to distinguish these context-dependent semantic shifts, resulting in over-generalization. The fourth error type is figurative and ironic language misinterpretation. The model lacks the capacity to recognize sarcasm, humor, or hyperbole, defaulting to literal interpretations. For example, the utterance "How many bones do you want, I have stock" carries an obvious humorous tone yet was classified as drug-related. Addressing this class of errors necessitates deeper pragmatic understanding, representing a promising direction for the integration of large pre-trained language models in future work.

 doi:10.20944/preprints202605.0904.v1

Overall, the identified error patterns suggest that while TextCNN excels at local pattern recognition in short texts, its performance is constrained by limited contextual modeling capacity and sensitivity to low-frequency or culturally specific expressions. These findings provide concrete guidance for model refinement, including data augmentation for rare coded terms, context-aware feature integration, and the incorporation of pragmatic reasoning capabilities.

4.5. Generalization Analysis

To assess the generalization capability and robustness of the proposed model under real-world conditions, two representative social media platforms were selected for cross-platform evaluation: Weibo, representative of short-text content, and WeChat Official Accounts, representative of long-form articles. Performance metrics for each platform are reported in Table 10.

As shown in Table 11, the model maintains consistently high performance across both platforms, with accuracy differing by less than 0.1% between the two settings. On Weibo short texts, the model achieves an F1-score of 0.9983, while on WeChat long-form content, it attains an F1-score of 0.9978. This marginal performance gap suggests that the multi-scale convolutional feature extraction mechanism of TextCNN is effective not only at capturing concise coded expressions in short texts but also at localizing semantically relevant patterns within longer and more complex documents. These results demonstrate satisfactory cross-platform adaptability of the proposed approach.

Table 11. Model performance comparison across different platform content types.

Platform Type	Accuracy	Precision	Recall	F1-Score
Weibo (short-text)	**0.9983**	**0.9982**	**0.9984**	**0.9983**
WeChat Official Accounts (long-text)	0.9978	0.9979	0.9977	0.9978

Temporal Robustness Analysis. Given the dynamic and rapidly evolving nature of drug-related coded language on social media, a three-month longitudinal evaluation was conducted to examine temporal performance degradation. As illustrated in Figure 9, the model sustains strong performance during the first month, with an accuracy of approximately 0.998. However, a gradual decline is observed in subsequent months, with accuracy dropping to 0.986 in the second month and further to 0.963 by the third month.

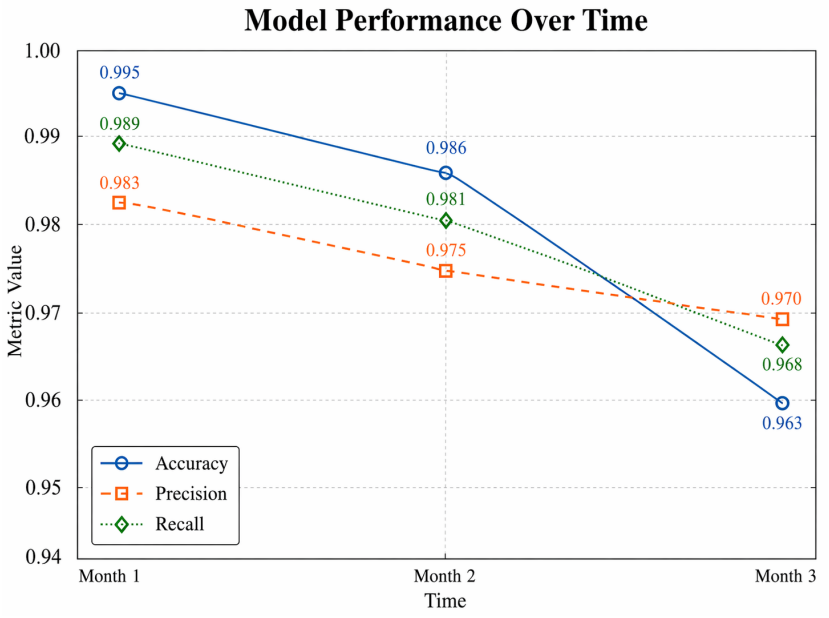

Figure 9. Performance trend chart of the model over three months.

 doi:10.20944/preprints202605.0904.v1

This performance degradation is attributable to the inherent linguistic drift of drug-related coded expressions: criminal communities continuously coin new slang terms or reassign novel meanings to existing vocabulary, a phenomenon that the static training corpus is unable to anticipate. As the temporal gap between training data collection and deployment widens, the model's coverage of emerging coded expressions diminishes accordingly. These findings highlight the necessity of periodic model retraining with updated data and suggest that incorporating online learning or continual learning mechanisms could be a promising direction for maintaining detection efficacy over time.

4.6. Summary

This section presented a comprehensive experimental evaluation of the proposed TextCNN-based model for drug-related coded language detection. The model achieves an accuracy, recall, and F1-score of 0.9981 on the test set, with an AUC of 1.0000, demonstrating strong and consistent detection performance. Comparative experiments against BERT-Base-Chinese, Chinese-RoBERTa-wwm-ext, SVM, and Naive Bayes confirm that TextCNN outperforms both large pre-trained language models and conventional machine learning baselines on this task, validating its suitability for short-text coded language recognition driven by local semantic pattern matching.

Error case analysis identifies four principal failure modes: (1) recognition failure for low-frequency or regionally specific coded terms; (2) context-induced misclassification of lexically ambiguous expressions; (3) difficulty in interpreting culturally or community-specific language; and (4) inability to recognize figurative, ironic, or humorous utterances. These findings provide concrete guidance for targeted model improvement in future work.

Generalization experiments across Weibo and WeChat platforms confirm that the model maintains stable performance in both short- and long-text scenarios, with a cross-platform accuracy difference of less than 0.1%. However, the longitudinal evaluation reveals a gradual performance decline over three months, attributable to the continuous linguistic evolution of drug-related coded expressions. This underscores the need for periodic model retraining and motivates future exploration of continual learning strategies to sustain long-term detection efficacy.

5. Conclusions

This study addressed the challenging problem of drug-related coded language detection on social media platforms by proposing a TextCNN-based intelligent recognition framework. Three primary contributions were made.

First, a dedicated dataset of 10,000 annotated samples was constructed through a multi-source collection strategy, integrating judicial documents from the China Judgment Online platform and anti-drug case reports from the Ministry of Public Security, supplemented by publicly available text crawled from Weibo, WeChat Official Accounts, and Douyin, as well as synthetically generated samples derived from a coded language lexicon. A double-blind annotation protocol with cross-validation achieved an inter-annotator agreement of $K = 0.91$, confirming high labeling reliability.

Second, a TextCNN-based classification model was designed with three parallel convolutional filter sizes (3, 4, 5) to capture local n-gram semantic features at multiple granularities. Global max-pooling was applied to aggregate salient features, followed by a fully connected layer with Softmax output. Input sequences were standardized to 100 tokens, initialized with 100-dimensional pre-trained Word2Vec embeddings fine-tuned during training, and optimized using the Adam optimizer with cross-entropy loss.

Third, comprehensive experiments on the constructed dataset demonstrated that the proposed model achieves accuracy, precision, recall, and F1-score of 0.9981, with an AUC of 1.0000, outperforming BERT-Base-Chinese, Chinese-RoBERTa-wwm-ext, SVM, and Naive Bayes across all evaluated metrics. Cross-platform generalization experiments confirmed stable performance in both short- and long-text scenarios, while longitudinal evaluation revealed gradual performance degradation attributable to the continuous linguistic evolution of coded drug expressions.

 doi:10.20944/preprints202605.0904.v1

Despite these promising results, several limitations remain. The model exhibits reduced sensitivity to low-frequency or regionally specific coded terms, struggles with lexically ambiguous and culturally specific expressions, and lacks the capacity to interpret figurative or ironic language. Future work will proceed along three directions. First, the dataset will be expanded through broader multi-platform and multi-regional data collection, incorporating crowdsourced annotation pipelines and a dynamically maintained coded language lexicon to improve coverage of rare and emergent slang. Second, fine-tuned pre-trained language models such as BERT , RoBERTa, and ERNIE [31] will be investigated to compensate for TextCNN's limitations in complex contextual reasoning; hybrid architectures combining convolutional local feature extraction with Transformer-based global semantic modeling will also be explored. Third, incremental and online learning strategies will be developed to enable continuous model updating without full retraining, supported by periodic data refresh cycles to sustain sensitivity to newly emerging coded expressions in evolving social media environments.

Author Contributions: Conceptualization, T.L and Y.D.; methodology, T.L.; software, X.Y.Y.; validation, T.L., Y.D. and X.Y.Y.; formal analysis, T.L.; investigation, X.Y.Y.; resources, Y.D.; data curation, X.Y.Y; writing—original draft preparation, T.L.; writing—review and editing, T.L.; visualization, X.Y.Y; supervision, Y.D.; project administration, Y.D.; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key R&D Projects of Sichuan Science and Technology Program OF FUNDER grant number 2024YFFK0123.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin, S. Y., Chien, S. Y., Chen, Y. Z., et al. Combating Online Malicious Behavior: Integrating Machine Learning and Deep Learning Methods for Harmful News and Toxic Comments. *Information Systems Frontiers* **2024**, *1*, 1–16.
2. Khan, U., Khan, S., Rizwan, A., et al. Aggression detection in social media from textual data using deep learning models. *Applied Sciences* **2022**, *12*(10), 5083.
3. Nahar, S., Any, O. H., Afrin, M., et al. Understanding of Drug Addiction Drug Abuse and Popular Drug Slang: A Narrative Review. *Journal of National Institute of Neurosciences Bangladesh* **2022**, *8*(1), 84–89.
4. Hu, C., Yin, M., Liu, B., et al. Detection of Illicit Drug Trafficking Events on Instagram: A Deep Multimodal Multilabel Learning Approach. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)* **2021**, 3838–3846.
5. Mackey, T. K., Kalyanam, J., Katsuki, T., et al. Solution to Detect, Classify, and Report Illicit Online Marketing and Sales of Controlled Substances via Twitter: Using Machine Learning and Web Forensics to Combat Digital Opioid Access. *Journal of Medical Internet Research* **2018**, *20*(4), e10029.
6. Sundaram, A., Subramaniam, H., Ab Hamid, S. H., et al. A Systematic Literature Review on Social Media Slang Analytics in Contemporary Discourse. *IEEE Access* **2023**, *11*, 132457–132471.
7. Liu, S., Gui, D. Y., Zuo, Y., et al. Good Slang or Bad Slang? Embedding Internet Slang in Persuasive Advertising. *Frontiers in Psychology* **2019**, *10*, 1251.
8. Holbrook, E., Wiskur, B., Nagykaldi, Z. Discovering Drug Slang on Social Media: A Word2Vec Approach with Reddit Data. **2024**.
9. Deng, J., Sun, X., Liu, J., et al. COLD: A Benchmark for Chinese Offensive Language Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022* **2022**, 3086–3100.
10. Ren, R., Zhao, J., Sun, X., et al. NLP-Based Review for Toxic Comment Detection Tailored to the Chinese Cyberspace. *IEEE Transactions on Computational Social Systems* **2025**.
11. Wu, L., Morstatter, F., Liu, H. SlangSD: Building, Expanding and Using a Sentiment Dictionary of Slang Words for Short-Text Sentiment Classification. *Language Resources and Evaluation* **2018**, *52*(3), 839–852.
12. Gadusu, S. R., McGinty, H. Semantic Similarity for Drug Slang Identification: A Comparative Analysis of Word2Vec and BERT. In *Proceedings of the 13th Knowledge Capture Conference 2025* **2025**, 190–193.

 doi:10.20944/preprints202605.0904.v1

13. Asim, M., Waqar, M., Alam, I. A Comparative Analysis of Deep Learning Methods for Slang Detection in Twitter Data. *Spectrum of Engineering Sciences* **2025**, *3*(12), 254–270.
14. Minaee, S., Kalchbrenner, N., Cambria, E., et al. Deep Learning-Based Text Classification: A Comprehensive Review. *ACM Computing Surveys* **2021**, *54*(3), 1–40.
15. Li, Q., Peng, H., Li, J., et al. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology* **2022**, *13*(2), 1–41.
16. Devlin, J., Chang, M. W., Lee, K., et al. 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 (NAACL-HLT)* **2019**, 4171–4186.
17. Hu, H., Phan, N., Geller, J., et al. An Ensemble Deep Learning Model for Drug Abuse Detection in Sparse Twitter-Sphere. In *Proceedings of MedInfo* **2019**, 163–167.
18. Tassone, J., Yan, P., Simpson, M., et al. Utilizing Deep Learning and Graph Mining to Identify Drug Use on Twitter Data. *BMC Medical Informatics and Decision Making* **2020**, *20*(Suppl 11), 304.
19. Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* **2014**, 1746–1751.
20. Soni, S., Chouhan, S. S., Rathore, S. S. TextConvoNet: A Convolutional Neural Network Based Architecture for Text Classification. *Applied Intelligence* **2023**, *53*(11), 14249–14268.
21. Shiozawa, K., Hayashi, H., Akiyama, S., et al. Detecting Slang on the Dark Web Based on Word Co-Occurrence Relationships in Anchor Texts. In *Proceedings of the 2026 40th International Conference on Information Networking (ICOIN)* **2026**, 347–352.
22. Hossain, N., Tran, T. T. T., Kautz, H. Discovering Political Slang in Readers' Comments. In *Proceedings of the International AAAI Conference on Web and Social Media* **2018**, *12*(1).
23. Sun, Z., Hu, Q., Gupta, R., et al. Toward Informal Language Processing: Knowledge of Slang in Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* **2024**, 1683–1701.
24. Carpenter, K. A., Samori, I. A., Kiang, M. V., et al. Large Language Models Can Disambiguate Opioid Slang on Social Media. *arXiv preprint arXiv:2603.10313* **2026**.
25. Patel, L., Alsobeh, A. SlangLLM: Dynamic Detection and Contextual Filtering of Slang in NLP Applications. In *Proceedings of the 2025 1st International Conference on Secure IoT, Assured and Trusted Computing (SATC)* **2025**, 1–6.
26. Aloraini, A. M., Batista-Navarro, R. T., Nenadic, G., et al. The SlangTrack Dataset: Supporting the Detection of Words Used in Slang Senses. In *Proceedings of the 6th International Workshop on Computational Approaches to Language Change (LChange'26)* **2026**, 1–19.
27. Hu, C., Liu, B., Ye, Y., et al. Fine-Grained Classification of Drug Trafficking Based on Instagram Hashtags. *Decision Support Systems* **2023**, *165*, 113896.
28. Wei, J., Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* **2019**, 6383–6389.
29. Mikolov, T., Sutskever, I., Chen, K., et al. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)* **2013**, 26, 3111–3119.
30. Cui, Y., Che, W., Liu, T., et al. Pre-Training with Whole Word Masking for Chinese BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2021**, *29*, 3504–3514.
31. Zhang, Z., Han, X., Liu, Z., et al. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)* **2019**, 1441–1451.