A Toxic Euphemism Detection Framework for Online Social Network Based on Semantic Contrastive Learning and Dual Channel Knowledge Augmentation

ARTICLE INFO

Keywords:
Toxic euphemisms
Contrastive learning
Implicit toxic
Knowledge fusion
Content moderation

ABSTRACT

For real-time content moderation systems, detecting toxic euphemisms remains a significant challenge due to the lack of available annotated datasets and the ability to deeply identify euphemistic toxicity. In this paper, we proposed the TED-SCL framework (Toxic Euphemism Detection based on Semantic Contrastive Learning) to solve these problems. Firstly, we collected nearly 8 million comments and constructed a toxic euphemism dataset (TE-Dataset), which contains 18,971 comments, covering six topics and 424 PTETs (Potential Toxic Euphemism Terms). Next, we employed contrastive learning to separate toxic euphemism samples from harmless ones in semantic space and enhance the model's ability to capture subtle differences. Lastly, we utilized a dual channel knowledge augmentation module to integrate background knowledge with toxic comments and improve the identification of toxic euphemisms. Experimental results demonstrate that TED-SCL outperforms existing SOTA in toxic euphemism detection tasks, achieving accuracy of 93.94%, recall of 93.36%, and F1 score of 93.23%. Furthermore, TED-SCL demonstrates better generalization, zero-shot capability, and greater robustness on different topics and datasets, which provides a new way for real-time content moderation systems to detect euphemistic and implicit toxicity effectively.

1. Introduction

Euphemisms (Zhu, Gong and Bansal, 2021) could conceal hateful and taboo meanings restricted by social rules. Therefore, many social media users use toxic euphemisms and spread illegal content or attack other groups publicly (Madaan, Setlur and et al., 2020). Bilibili[1] is a video-sharing website with a user base of more than 65% among young people in China. The website currently has over 220 million official members and more than 300 million active users each month. Malicious users on Bilibili create toxic euphemisms and spread inappropriate content, including offensive language and discriminatory bias. For example, "橄榄" (Olive) implies "干烂" (Rotten), "默" (Silence) refers to "黑人" (black people), and "蝈蝻" (Cricket) means "中国男人" (Chinese men).

Toxic euphemism comments on social networks evade traditional detection methods based on ban lists because of their inherently deceptive nature. On the one hand, toxic euphemisms contain harmful underlying meanings but appear to be ordinary words. On the other hand, traditional methods lack background knowledge of PTETs (Potential Toxic Euphemism Terms), resulting in a high rate of false negatives. For example, if a toxic euphemism comment containing the PTET "橄榄" (Olive) appears in a innocuous context, the toxicity classifier may incorrectly classify the comment as non-toxic. Existing euphemism studies mainly focus on euphemism term extraction (Chaves and Gerosa, 2021) and identification (Hu, Li and Wu, 2024b), which are mainly divided into two sub-tasks: potential euphemism term (PET) extraction and euphemism term mapping (Felt and Riloff, 2020). The PET extraction typically involves comparing the vector similarity of identical phrases or expressions in different corpora to identify potential euphemism phrases. Meanwhile, the Euphemism term mapping relies on algorithms, such as expanding seed word sets, calculating semantic similarity, and performing sentiment analysis to infer the underlying meaning of euphemisms. Due to the lack of large-scale (Zhu et al., 2021; Keh, Bharadwaj and Liu, 2022) and high-quality annotated datasets, existing methods mostly rely on unsupervised algorithms. Based on our findings and existing research, there are several main challenges that remain in the toxic euphemism detection:

- *Dataset limitations.* Currently, there is a significant shortage of large-scale and annotated datasets for toxic euphemism detection. The related existing toxic euphemism comment datasets are often small and focused on

*Corresponding author
ORCID(s):
[1] https://www.bilibili.com/

limited topics (Gavidia, Lee and et al., 2022). While larger and more diverse toxic comment datasets exist, they lack sufficient toxic euphemism terms and specific annotations for those terms. Therefore, there is an urgent need to create large-scale, high-quality annotated datasets for toxic euphemism detection.

- *Feature vector collapse.* Because toxic euphemisms comments have subtle semantic differences, the embedding vectors of toxic euphemisms comments collapse in the vectors space, which makes it hard for detection models to classify the toxic and healthy comments.

- *Toxic euphemisms comprehension.* Toxic euphemisms related to specific topics have different background information including culture and history. So the toxic euphemisms detection models need to integrate external toxic euphemisms knowledge and understand the underlying meanings of euphemisms appropriately in different contexts.

To solve these challenges, we proposed a novel method for detecting toxic euphemisms on social networks. Our proposed method can detect toxic euphemism comments on social networks, providing a new way for the research of toxic euphemism detection. Firstly, we collected a corpus of toxic euphemism comments from Bilibili platform and constructed a Chinese toxic euphemism comment dataset (TE-Dataset). Secondly, we proposed the TED-SCL framework, which represents **T**oxic **E**uphemism **D**etection based on **S**emantic **C**ontrastive **L**earning. Finally, through several experiments, we demonstrated that TED-SCL outperforms existing baselines.

- We constructed the first Chinese toxic euphemism comment dataset (TE-Dataset) by collecting millions of video comments and bullet screen messages from the Bilibili platform. Additionally, we developed an annotated dictionary of euphemistic paraphrases. This dataset provides a foundational resource for future research on toxic euphemisms, significantly advancing the study of nuanced toxic language in social contexts.

- We proposed the TED-SCL framework, which leverages enhanced semantic contrastive learning to improve the representation of toxic euphemisms. By dispersing toxic euphemism comment vectors in the semantic space through three innovative data augmentation techniques, the framework achieves a more uniform representation distribution, leading to a detection accuracy of 93.94%.

- Our dual-channel knowledge augmentation module integrates background knowledge of toxic euphemisms to enhance model understanding. By employing shared attention layers, the module effectively captures weighted information and fuses euphemism-specific background knowledge. This combination of dual-channel augmentation and semantic contrastive learning significantly improves the model's generalization, robustness, and applicability to tasks involving toxic and euphemistic language understanding.

By adaptively integrating contrastive learning with dual-channel augmentation, our framework establishes a novel paradigm for addressing the challenges of detecting euphemistic and toxic language. This innovative approach not only deepens the understanding of toxic euphemisms but also lays a theoretical foundation that can be extended to other NLP tasks requiring sophisticated language interpretation.

This paper is organized as follows. Section 1 summarizes the existing situation of toxic euphemisms on social networks, discusses the challenges of toxic euphemism detection, and summarizes the work contributions of this paper. Section 2 provides a comprehensive review of research related to toxic euphemism detection, including slang tracking, dark jargon, euphemism recognition, and toxic language detection. Section 3 details the construction of the dataset and the methods. Section 4 describes the experimental procedures and results. Section 5 analyzes the contributions and limitations of this work. We have released our code and TE-dataset on https://github.com/yiyepianzhounc/TED-SCL.

2. Related Work

The study of toxic euphemisms is a relatively new field with directly relevant research being limited. In terms of how they are used and spread, Internet slang on social networks is similar to toxic euphemisms. In research methodology, the mining and analyzing of dark jargon and the recognition of euphemisms resemble toxic euphemisms. In terms of the detection task, toxic language detection aligns with toxic euphemisms. Therefore, we provide a comprehensive summary of these related research efforts.

2.1. Slang Tracking

In the work of tracking Internet slang, researchers have employed multi-layer perceptrons and character-based convolutional embedding method (Pei, Sun and Xu, 2019) to achieve automatic detection of slang. However, this method could not ascertain the underlying meaning of slang well. To solve the problem, a topic-based method (Matsumoto, Ren and Matsuoka, 2019) has been proposed and achieves slang identification with high accuracy. More studies have demonstrated the superiority of pre-trained models on social media data (Sun, Zemel and Xu, 2021). Specifically, fine-tuned IndoBERT models (Fernandez, Winata and Fasya, 2022) can roughly classify slang as positive, neutral, and negative. In the construction of slang dictionaries, LIWC-UD (Bahgat, Wilson and Magdy, 2022) encompasses many common slang used on public platforms like Twitter. Moreover, Kravchenko et al. (2023) provided definitions of slang and developed a telegram chatbot with an Internet slang dictionary. To trace the slang, Sun et al. (2021) proposed a framework for simulating vocabulary selection by speakers in slang contexts.

The current research on Internet slang is insufficient in terms of semantic recognition and need to improve the accuracy. Existing methods often failed to capture the nuanced meanings and evolving nature of slang.

2.2. Dark Jargon

The utilization of Word2vec and LDA clustering methods in the early stages (Zhao, Zhang and et al., 2016) has proven beneficial for the extraction and comprehension of cryptic terminologies prevalent in cybercriminal activities. Additionally, the Keyword Detection and Expansion System (KDES) (Yang, Ma and et al., 2017) is anchored on China's leading Baidu search engine. The KDES has identified over 400,000 dark jargons through an analysis of search results linked to common terms. Furthermore, automated methodologies (Portnoff, Afroz and et al., 2017) scrutinize underground forums utilizing information extraction and named entity recognition techniques and these methodologies heavily rely on dataset quality.

Existing unsupervised methods for dark jargon identification primarily rely on comparing across corpora and enhanced word embeddings. Yuan, Lu and et al. (2018) employed improved word2vec embeddings to train word vectors and identified potential dark jargon by detecting candidate word sets that significantly differ between background and target corpora. To enhance accuracy in recognizing dark jargon, Lao, Zhang and et al. (2021) introduced a topic relevance coefficient for dark jargon vocabulary and integrated it with contextual information. To analyze Chinese dark jargon and improve accuracy, DC-BERT (Ke, Chen and Wang, 2022) employed BERT-base pre-trained models (Devlin, Kenton and Toutanova, 2019) to represent word vectors, demonstrating effective dark jargon detection on the dark web. Additionally, compound dark jargon with criminal intent (Hada, Sei and et al., 2023) can be detected based on differences in vocabulary usage in Internet posts. Some research has also explored supervised learning-based methods for dark jargon detection. SICM (Wang, Su and et al., 2022) employed a sequence labeling approach and integrated speech, character, and lexical features into the lower layers of BERT via the attention layer. The CJI-Framework (Wang, Hou and Wang, 2021) utilizes transfer learning and the vector projection method to identify dark jargons, which introduces seven new features based on vectors, morphology, and lexicons to improve the accuracy in recognizing dark jargons.

However, the semantic recognition accuracy of dark jargon detection remains insufficient (Hou, Wang and Wang, 2022), particularly for new euphemistic jargons. Additionally, existing research predominantly focuses on non-public platforms such as hacker forums, Telegram, and the dark web, But researches on public platforms that could have more underlying dark jargons and euphemisms are limited.

2.3. Euphemism Recognition

Euphemism plays an important role in daily dialogue and social interaction (Rababah, 2014), such as political discourse and doctor-patient interaction (Kapron-King and Xu, 2021). Some euphemisms are employed to describe criminal activities such as drugs, prostitution, and gun smuggling (Hu et al., 2024b), which is similar to dark jargons.

To coarsely classify euphemisms, Felt and Riloff (2020) investigated the classification of euphemisms for three themes: shooting, lying and stealing. To detect and classify euphemisms at a finer granularity, Zhu et al. (2021) treated euphemism recognition as a fill-in-the-blank task and used a masked language model combined with self-supervised learning. To reduce the influence of low-quality datasets, Keh et al. (2022) used a KNN-based model by filtering the PET (potentially euphemistic term) dataset manually and using BERT-Score similarity (Gavidia et al., 2022). Frenda, Cignarella and Basile (2022) proposed a language-driven proof-of-concept method to distinguish potential euphemisms conceptually, which found more PETs. To avoid semantic ambiguity in euphemisms, Kesen, Erdem and et al. (2022) combined textual descriptions and visual images through images generated by euphemisms. In addition,

some experimental results (Wiriyathammabhum, 2022) show that adding more word vector channels does not simply improve model performance.

2.4. Toxic Language Detection

The toxicity of comment on social networks includes explicit and implicit toxicity, and the detection task of implicit toxic language and toxic euphemisms both classify comments as toxic and non-toxic. Detecting toxic comment on social networks has related to the toxic euphemisms detection on the contexts (Eronen, Ptaszynski, Masui, Arata, Leliwa and Wroczynski, 2022), but the detection tasks of toxic euphemisms are focused on more euphemistic and challenging contexts (Fortuna, Soler-Company and Wanner, 2021).

Explicit Toxic Language. Several traditional methods have been employed to address problems of explicit toxic language detection (Aljawazeri and Jasim, 2024). Sequence labeling methods and data-driven convolutional neural network (Elbasani and Kim, 2022) have proven effective in identifying online toxic language. The Multitask learning has made progress in the implicit toxic language detection task in rencent research. Ameya, Feng and Yue (2020) analyzed the effectiveness of multi-task learning (MTL) in reducing bias in toxic comment detection, and proposed an approach that addresses identity bias by leveraging shared and specialized task representations. Moreover, Jia, Wu and et al. (2023) utilized a slot-filling approach to capture global information between different semantic components. Based on MTL (Multi-task learning) framework, Gupta, Lee and De-Arteaga (2023) enables models to specialize in relationships pertinent to each demographic group while leveraging shared properties across groups and enhanced the recall of the baseline model. To improve the performance of traditional MTL, A new approach aimed at mitigating negative transfer issues is proposed. And the method is based on task-awareness (de Paula, Rosso and Spina, 2023) and addresses the performance degradation from the sharing of noisy information across multiple tasks.

Implicit Toxic Language. Implicit toxic language detection is a significant challenge in NLP. Jiawen, Zhuang and Hao (2023) developed the AugCOLD dataset and used multi-teacher knowledge distillation to enhance unsupervised detection. Anjalie and Yulia (2020) applied propensity matching and adversarial learning to detect implicit gender bias in comments. However, traditional explicit methods alone are insufficient for implicit toxic language. Jessica (2022) introduced entity linking to improve detection, showing its benefit for explicit toxic language, but limited impact on implicit toxic language. To fully consider the impact of social relationship information on the detection of implicit toxicity, Sreyan, Manan and Purva (2023) proposed CoSyn that models dialogue context and social relationships on social networks for implicit hate speech detection. To reduce the reliance on large-scale datasets, researchers have proposed various approaches to enhance detection. Youngwook, Shinwoo and Yo-Sub (2022) introduced a framework using minimal labeled examples and TrackIn to enhance implicit toxic language detection without large datasets. Meanwhile, Xiaochuang and Yulia (2020) developed ImpCon, a contrastive learning method to improve cross-dataset performance, boosting model generalization.

In summary, the existing research has the following deficiencies. (1) detecting toxic euphemisms remains relatively under explored and there is limitation on the datasets of toxic euphemism detection. Existing datasets of euphemisms recognition and dark jargons detection, proposed to research euphemistic terms extraction and identification, which are not suitable for detecting toxic euphemism comments on social networks. (2) Existing toxic language methods based on static word embeddings and pre-trained models failed to detect toxic euphemisms, which does not fuse the external knowledge of toxic euphemisms and lacks of euphemistic meaning comprehension. Therefore, we proposed a new framework based on contrastive learning and dual channel knowledge augmentation, which obtains uniform features of the toxic euphemisms, fuses external knowledge of euphemisms and effectively detects toxic euphemisms comments.

2.5. Research Objectives

Based on the existing research limitations, our research objectives of toxic euphemism detection are to establish necessary research resources and provide an effective end-to-end detection method. We summarized the research Objectives of toxic euphemism detection from three different aspects: methodologies, affected factors and model reliability: 1) What is the most effective methodology and algorithm for detecting toxic euphemisms through NLP techniques? 2) How does deep background knowledge or sentiment signals affect the detection performance? 3) How can we ensure the reliability of detection models, including minimizing false positives and false negatives? Firstly, we defined the PTET (Potential Toxic Euphemism Term) and constructed the TE-Dataset and PTETD (Potential Toxic Euphemism Terms Dictionary), and then proposed the TED-SCL framework. Secondly, we performed extensive comparative experiments to evaluate both traditional methods and detection approaches based on the large language

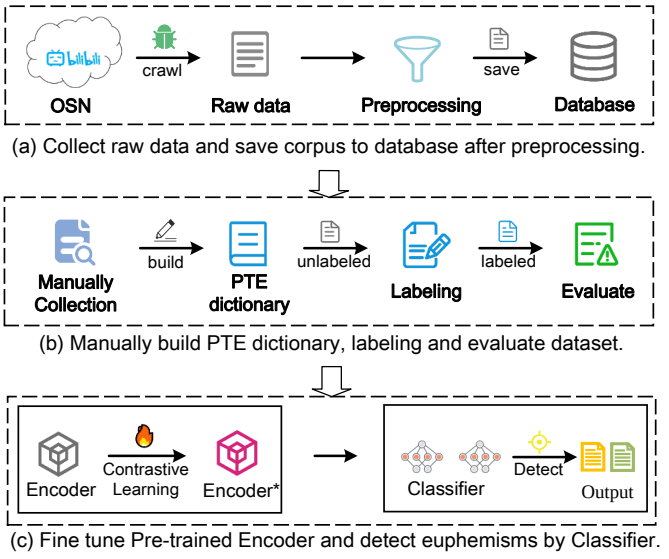


Figure 1: Overview of TED-SCL framework. Step (a) involves collecting a substantial amount of data related to toxic euphemisms from various sources. Step (b) processes this data and manually annotates it to create a labeled dataset. Step (c) uses the enhanced encoder and classifier to train the model, and then effectively classifies toxic euphemism comments.

models. Lastly, we conducted experiments to show the stability and reliability of our proposed method, including experiments of generalization and robustness.

3. Methods

This paper proposed the TED-SCL framework for detecting toxic euphemism comments on social networks, using semantic contrastive learning and dual-channel knowledge augmentation. The framework of our research includes three main steps, as shown in Figure 1: 1) We developed a web crawler using the Scrapy framework and collected raw data from the Bilibili platform. The seed videos were manually selected for various topics, and the web crawler collected the corresponding comments by each video's ID. 2) We filtered video comments from different channels on Bilibili, based on seed keywords from the search engine of Bilibili. Additionally, we annotated these comments and evaluated the consistency of labeling. 3) We designed the model architecture of TED-SCL, which has three main modules: a pre-trained RoBERTa encoder (Cui, Che, Liu, Qin, Wang and Hu, 2020), a contrastive learning mechanism, and a toxic euphemism classifier.

Firstly, video comments and barrage information from the Bilibili platform are collected using web crawlers. After that, the data is cleaned and filtered. Secondly, the TE-Dataset and PTETD are created by selecting relevant data and annotating it manually. The PTETD includes potential toxic euphemistic terms and their annotations. Then, initial vectors are generated and improved through representation extraction Module. Finally, a classifier is trained to detect toxic euphemisms, using a dual-channel knowledge enhancement module.

3.1. Dataset Construction

To address the lack of datasets for toxic euphemism comments, we selected Bilibili as the target platform, given the rapid proliferation and widespread use of toxic euphemisms on this site. To construct our corpus, we developed an efficient web crawler that uses the Scrapy framework to collect video comments. Detailed information on the construction of the TE-Dataset is provided in Table 1. The total average and minimum length of toxic euphemism comments in the TE-dataset are 33.7 and 3.33. In addition, the total number of PTETs in the TE-dataset is 424.

3.1.1. Collection Strategy

To investigate different toxic euphemisms on social networks, the dataset must cover a wide range of topics. The seed videos were manually selected for various topics, and a web crawler got the corresponding comments by each video's ID. To ensure diversity and balance of comments, seed videos were chosen from different channels on Bilibili,

Table 1
Details of the TE-Dataset. "PTE Terms Number" refers to the number of PTETs. "Tox." indicates the count of toxic euphemism comments, while "N-Tox." represents the non-toxic comments. "Total Number" represents the total number of comments in different topics, and "Average Length" represents the average length of all comments.

Topics	PTE Terms Number	Toxic euphemism Comments				
		Tox.	N-Tox.	Total Number	Average Length	Minimum Length
"种族" (Race)	244	2,649	2,381	5,030	36.2	4.0
"色情" (Sexism)	134	4,054	2,020	6,074	26.7	2.0
"日常" (General)	57	593	560	1,153	78.8	2.0
"性别" (Gender)	157	1,319	1,299	2,618	36.7	5.0
"性少数群体" (LGBT)	96	887	921	1,808	46.3	4.0
"地区" (Region)	81	1,066	1,222	2,288	11.2	3.0
Total	424	10,568	8,403	18,971	33.7	3.33

Table 2
The initial seed keywords. The Topic includes six categories of toxic euphemisms from different topics, and the seed keywords are collected manually. The unique phonetic and glyph characteristics of these Chinese phrases make it difficult to accurately translate them into English equivalents.

Topic	Initial Seed Keywords
"种族" (Race)	白皮, 棒子, 黑皮, 默, 狗, 虫混, 小黑, 猩猩, 黑女, 黑蛆, 媚黑
"色情" (Sexism)	菊花, 步兵, 骑兵, 社保, 擦边, 卖肉, 牛头人, 黄油, 开导, 超市, 金针菇, 烧鸡
"日常" (General)	白左, 小粉红, 鸡, 神友, 狗, 原, 兔友, 圣母, 批, 五毛, 孝子, 恶心, 弱智, 日杂, 猴子
"性别" (Gender)	女拳, 仙女, 幕刃, 普信, 打拳, 批, 鸡, 子宫, 恶心, 阿娜, 默, 下头, 拳师, pua, 奴隶
"性少数群体" (LGBT)	反同, 基佬, 男同, 恶心, txl, gay, 通讯录, 骚, 恐同, 同志, 染艾, 直佬
"地区" (Region)	偷井盖, 南满, 默, 南大人, 南蛮, 蛮, 棒子, 弯弯, 飞舟, 小日子, 飞周, 蛮夷

based on seed keywords from search engine of Bilibili. Video popularity was assessed using metrics such as view count, likes, rewards, comments, and shares. The initial seed keywords included six topics: "种族" (Race), "色情" (Sexism), "日常" (General), "性别" (Gender), "性少数群体" (LGBT), and "地区" (Region). Detailed information on initial seed keywords is provided in Table 2.

More details about the initial seed keywords are provided in Table 2. The dataset was collected between January 2020 and December 2021, and we used a breadth-first search algorithm for data collection. Firstly, the crawler collected comments from target videos and stored them in the database. Then, the crawler selects the top 10 stored videos with the highest recommendation ranks, calculated by video popularity. These videos become child nodes for the next traversal round, with duplicate nodes eliminated. After four traversal rounds, the crawler collected about 800 million comments from 5,000 videos. Based on the initial pool, we filtered approximately 100,000 comments using keywords and further refined the dataset by applying a filter for the minimum comment length. After manual annotation and multiple rounds of expert review, we finalized a balanced dataset of approximately 18,971 comments, containing the same number of positive and negative samples.

3.1.2. Data Annotation

To improve the accuracy of data annotation, this work follows several core principles for data labeling. All annotators must perform their tasks independently according to annotation principles. The consistency of annotation is verified by a third party after labeling, and the comments with inconsistent labels will be moved.

Annotator. On the one hand, the recognizing of annotators' identities and cultural background can influence their each judgment, so we selected graduate students from diverse fields to annotate the dataset at the same time. On the other hand, the understanding of euphemisms among individuals is different because of cultural background, so each specialized annotator was allowed to interpret the semantics of toxic euphemisms based on their own understanding. Additionally, annotators evaluated and labeled euphemisms independently. Finally, we conducted sampling checks

on the quality of annotation to ensure consistency and any significant discrepancies in annotated euphemisms were thoroughly reviewed and resolved by the annotation team.

Analysis. Toxic euphemism comments convey toxic information including race discrimination, children pornography and groups enmity by distorting the healthy ordinary words through techniques like variant words, metaphor, or sarcasm. Based on the analysis of dataset samples, the characteristics of toxic euphemistic comments can be summarized as follows: (i) inconsistency between the intended meaning of the comment and its literal interpretation, (ii) use of uncommon phrases or combinations of special characters and numbers, (iii) observable shifts in the tone or mood compared to the normal context, (iv) coherence of logic between comments is ridiculous.

Annotation. After construed a basic dictionary through prior knowledge and datasets analysis, we expanded PTETs list and annotated each term. PTET (Potential Toxic Euphemism Term) refers to a word or phrase that carries a potentially toxic meaning, but its toxicity depends on the specific context. The final PTETs dictionary compiled by us includes 424 PTETs within six topics. Annotators classified each comment as either toxic or normal by comprehension of their contextual information according to the **principles** of annotation, which are as follows: (i) If one comment including one or more PTETs can be reasonably sensed as harmless but not as offensive, the normal comment is labeled “0”, (ii) If one comment including one or more PTETs can reasonably be sensed as offensive meaning, but not as normal or healthy meaning, the toxic comment is labeled “1”, (iii) If one comment including PTETs can not be sensed as distinct meaning, the comment is labeled “-1”, (iv) The comments that label consistency is less than 0.4 or label are "-1" will be reevaluated by the committee of experts, unqualified comments will be removed from TE-Dataset.

Evaluation. For evaluation of data annotation, this paper assigned four annotators to independently review all comments of the TE-Dataset, ensuring the removal of redundant text and excessive repetition to enhance data annotation reliability. Finally, the consistency of annotation results is evaluated using the Kappa coefficient. The definition of the Kappa coefficient is shown in Equation (1), (2), (3):

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{1}$$

$$P(A) = \frac{\mid A \cap B \mid + \mid C \cap D \mid}{E} \tag{2}$$

$$P(E) = \frac{\mid A \mid \times \mid B \mid}{\mid E \mid^2} + \frac{\mid C \mid \times \mid D \mid}{\mid E \mid^2} \tag{3}$$

where A is the set of comments labeled by the first annotator, B is the set of comments labeled by the second annotator, C is the set of comments for which the first annotator could not tell if they were positive samples, D is the set of comments for which the second annotator could not tell if they were positive samples, E is the set of all comments, $|\cdot|$ is the size of a set. By calculation, the Kappa coefficient of dataset annotation was 0.81, and the results show that all annotators reached a relatively high consensus in classifying toxic euphemisms.

3.1.3. Dataset

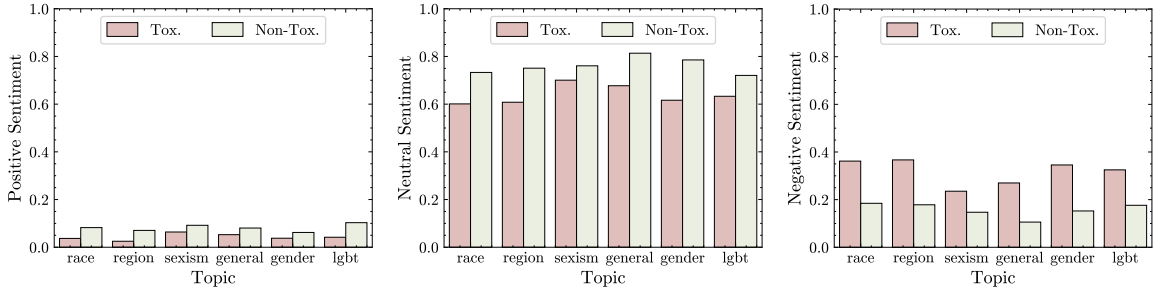
In summary, we combined existing datasets that contains PTETs (Lu, Xu and et al., 2023) to expand TE-Dataset. As shown in Table 1, the TE-Dataset was collected from comments of six topics and includes the PTETD of 424 PTETs. The TE-Dataset has 10,568 toxic euphemism comments containing PTETs and 8,403 non-toxic comments, with an average length of 33.7 characters. Figure 2 shows the results of sentiment analysis conducted on the TE-Dataset using the Tencent Cloud API (Wang, Huang and et al., 2023), where "Positive" is positive sentiment, "Negative" is negative sentiment, "Neutral" is neutral sentiment, "Non-Tox." represents non-toxic comments, and "Tox." represents toxic euphemisms. We observed that compared to non-toxic comments, toxic euphemism comments exhibited a slightly weaker average positive sentiment, a more intense average negative sentiment, and a very similar average neutral sentiment, indicating that sentiment features are one of the prominent characteristics of toxic euphemisms.

As research on toxic euphemism detection falls within the topic of toxic content detection, we compared the TE-Dataset with datasets for toxic language detection (Deng, Zhou and et al., 2022; Zhou, Deng and et al., 2022; Jiang, Yang, Liu and Zubiaga, 2022), as shown in Table 3. TOXICN (Lu et al., 2023) contains a portion of implicit toxic content that aligns with the definition of toxic euphemisms; therefore, the TE-Dataset incorporates toxic euphemisms

Table 3
Comparison of Chinese datasets of toxicity detection. "Features" refers to the annotated features in the dataset. "Tex.", "Eup.", and "Sen." are labels of annotated comments, euphemisms, and sentiment. The "Par." represents the label of euphemisms meaning. "Size" represents the total number of comments in each dataset. "Balance" represents the ratio of positive to negative samples.

Datasets	Features				Size	Balance(%)	Type	Language
	Txt.	Eup.	Emo.	Par.				
Jigsaw Toxic Comment Datase	✓				1,700,000	20.00	Diverse Toxicity	Multilingual
Hatexplain (Mathew et al., 2021)	✓				16,000	42.85	Hateful	Multilingual
OLID (Zampieri et al., 2019)	✓				14,000	52.10	Offensive	English
COLD (Deng et al., 2022)	✓				37,480	48.10	Offensive	Chinese
SWSR (Jiang et al., 2022)	✓				8,969	34.50	Hateful	Chinese
CDial-Bias-Utt (Zhou et al., 2022)	✓				13,394	18.90	Hateful	Chinese
CDial-Bias-Ctx (Zhou et al., 2022)	✓				15,013	25.90	Hateful	Chinese
TOXICN (Lu et al., 2023)	✓	✓			12,011	53.80	Offensive	Chinese
TE-Dataset (ours)	✓	✓	✓	✓	18,971	55.71	Euphemistic Toxicity	Chinese

from this dataset. In addition, the TE-dataset is different from traditional datasets like Jigsaw Toxic Comment Dataset[2], OLID (Zampieri et al., 2019), and HateXplain (Mathew et al., 2021) by focusing on euphemistic toxicity conveyed through PTETs, which heavily rely on deep external contextual information. While other datasets primarily concentrate on explicit forms of toxicity, the TE-datase offers a unique challenge for detecting toxic euphemism comments on social networks. Moreover, the TE-dataset includes 18,971 annotated comments of six different topics collected from Bilibili, and provides a diverse and context-rich resource for detecting toxic euphemisms.



(a) positive sentiment result (b) neutral sentiment result (c) negative sentiment result

Figure 2: Sentiment analysis of TE-Dataset. These charts illustrate the distribution of positive, neutral, and negative sentiment distribution of toxic euphemism comments and non-toxic comments in the TE-Dataset. The chart (a) shows the positive sentiment distribution in toxic and non-toxic comments in different six topics contain fewer positive sentiments overall. The chart (b) shows the neutral sentiment distribution, and non-toxic comments contain significantly more neutrality. The chart (c) shows the negative sentiment distribution, where toxic euphemism comments contain more negative sentiment.

3.2. Representation Extraction Module

In Figure 3, the model architecture of TED-SCL has three main parts: a pre-trained RoBERTa (Cui et al., 2020) encoder for generating initial vectors, a contrastive learning framework that uses different data augmentation techniques to improve the euphemism vectors, and a toxic euphemism classifier that combines multiple features for accurate classification. The next section explains the representation extraction process in TED-SCL in more detail.

[2] https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/overview

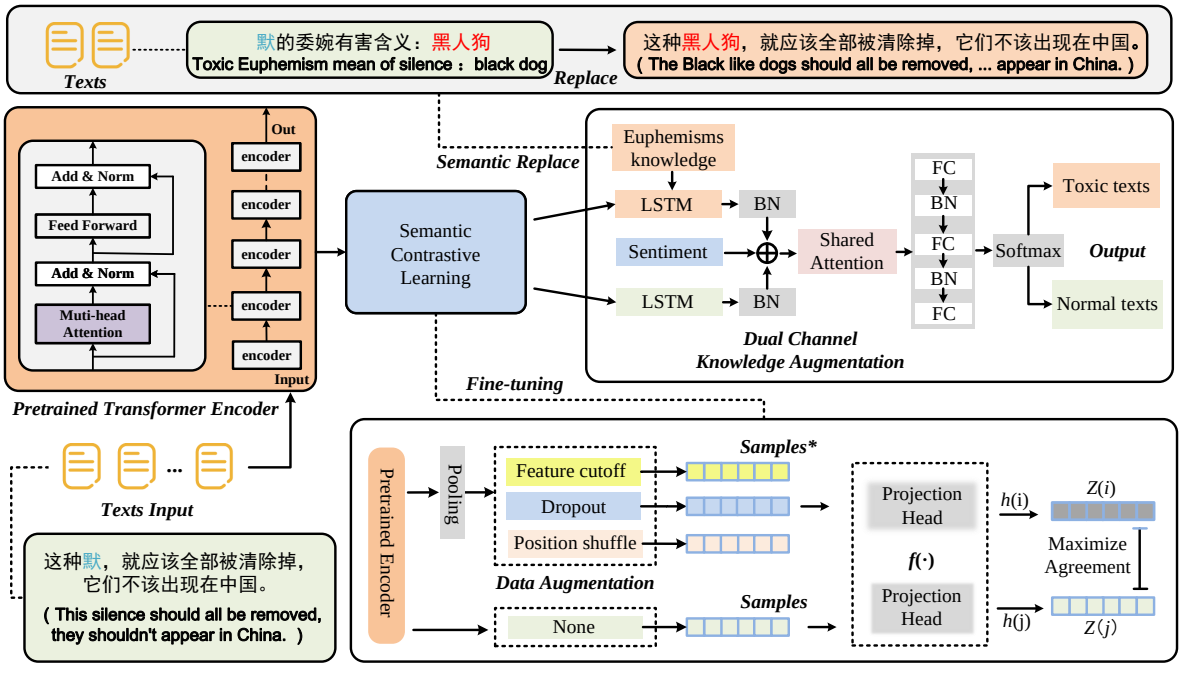


Figure 3: The model architecture of the TED-SCL framework consists of three main modules: a pre-trained encoder, a contrastive learning module, and a dual-channel knowledge enhancement module. The pre-trained encoder generates the initial embedding vector for the input comments. The contrastive learning module refines the encoder's output through contrastive training. The dual-channel knowledge enhancement module incorporates external lexical knowledge into the comment vectors.

3.2.1. Input Layer

Data preprocessing includes tasks such as removing web links, foreign characters, and special symbols. After data cleaning and selection, the annotated text sequences are input into the feature extraction layer of TED-SCL. As shown in Figure 1, the first part of the input layer takes preprocessed text data (video comments and video danmaku) $I = \{I_1, I_2, ..., I_n\}$ as input.

After preprocessing, each comment is tokenized into n tokens, where n is the predetermined max sequence length. Any part of the comment exceeding n tokens is truncated, and shorter sequences are padded with zeros. The core of multi-head attention mechanism is the combination of dot-product and matrix multiplication between each pair of tokens in the sequence. As a result, longer sequences can slow processing speeds and increase memory requirements. The optimal max sequence length is determined through hyperparameter experiments.

3.2.2. Encoder Layer

In the encoder layer, we employed an improved RoBERTa (Cui et al., 2020) model based on BERT (Wan, Pan and Yang, 2021; Zhang, Wu, Xu, Cao, Du and Psounis, 2024; Nelatoori and Kommanti, 2023). The pre-trained model (Gu, Luo and Yang, 2024) has shown remarkable performance across various natural language processing tasks in sentence-level tasks (Le, Pham and et al., 2024) and token-level tasks (Li, Sun and Han, 2020b). BERT (Devlin et al., 2019) is designed to acquire representations by considering context from both left and right directions across all layers. Therefore, fine-tuning pre-trained BERT with an additional output layer enables adaptation to specific tasks without substantial modifications to the model structure.

Specifically, the pre-processed text $\{I_1, I_2, ..., I_n\}$ is tokenized and then fed into the encoder layer for word embedding extraction. Subsequently, through average pooling, the obtained word embeddings are transformed into high-quality semantic features $W = \{W_1, W_2, ..., W_n\}$, where each W_i is 768 or 1024 dimensional vector. The calculation is performed as follows:

$$I' = Tokenizer(I) \tag{4}$$

$$W = Encoder(I^{'}) \tag{5}$$

$$S = AveragePooling(W_1, W_2, ..., W_n) \tag{6}$$

where S is the initial representation vector of potential toxic euphemisms.

3.2.3. Data Augmentation

The contrastive learning framework (Zeng and Cui, 2022) generates sample pairs through data augmentation strategies, then feeds all sample pairs of the same batch into the contrastive loss function to calculate the contrastive loss. Data augmentation methods mainly including implicit types (adversarial attacks (Zhang, Benz, Imtiaz and Kweon, 2020), cutoff (Chen, Shen, Chen and Yang, 2021), etc.) and explicit types (token deletion, token replacement, token shuffling, etc.). Based on experimental results, our work used comprehensive methods including token shuffling, cutoff, and RID (random inactivation dropout) and improved the performance of proposed method. besides them, we construct comparable samples by replacing PTET with annotations, which is called replacing operation in Figure 4.

Token Shuffling. Token shuffling is to perturb the original order of the input comment by manipulating its position and comment embedding. Transformer senses the sequence of comments through position embeddings, so we can shuffle the comments h_i, h_{sep} and h_{cls} of the sequence $C = [h_{cls}, h_0, ..., h_{N-1}, h_{sep}]$ to complete the sequencing reconstruction of the original comment (Lee, Hudson, Lee and Manning, 2020).

Cutoff. For the sequence $X = [x_1, x_2, ..., x_L]$, the embedding matrix is denoted by $\boldsymbol{W} \in \mathbb{R}^{L \times d}$, where $\omega_{i,j}$ is the j dimension of the embedding vector corresponding to the i token, and d is the dimension of the input embedding. Partial samples can be obtained by cutting the vector along any dimension, and the proposed method is called Cutoff. Meanwhile, we avoid disrupting the order of toxic euphemism words by masked the whole toxic euphemisms.

RID. Dropout is one widely used regularization method to avoid overfitting (Zhang and Xu, 2024) and also effective as a reinforcement strategy for contrastive learning. Dropout randomly zeroes each element in the embedding layer with a specific probability. This strategy is different from Cutoff because each element is considered separately.

3.2.4. Contrastive Loss

The contrastive loss function is defined as a contrastive prediction task. Given a set $\{\bar{x}_k\}$ consisting of a pair of positive samples $\bar{\mathbf{x}}_a$ and $\tilde{x}_b$, the contrastive prediction task aims to identify $\{\bar{x}_k\}$ for a given $\bar{\mathbf{x}}_a$ in $\{\bar{x}_k\}_{k \neq a}$. Two separate data augmentation operators are sampled from the same augmentation set ($t \sim \mathrm{T}$ and $t' - \mathrm{T}$) and applied to each data sample to obtain two correlated sample pairs. Then, using the resulting sample pairs, the basic encoder network $f(\cdot)$ and the projected neural network $\mathbf{g}(\cdot)$ are trained to maximize consistency using contrast loss. After the completion of training, the projection head $\mathbf{g}(\cdot)$ is removed, and the encoder $f(\cdot)$ along with the representation $\mathbf{h}$ are utilized for downstream tasks.

Specifically, by randomly sampling $2N$ batches of samples, N data sample pairs are obtained. Then the $2(N-1)$ augmented sample pairs other than the positive sample pair in the same batch are considered as negative samples. The $sim(u, v) = u^{\mathrm{T}}v/||u||||v||$ is used to represent the cosine similarity between u and v after regularization by $\boldsymbol{l}_2$. Thus, the definition of the loss function of positive sample pair to (a, c) is as follows :

$$\ell_{a,c} = -\log \frac{\exp(sim(z_a, z_c))}{\sum_{k=1}^{2N} 1_{[k \neq a]} \exp(sim(z_a, z_k))} \tag{7}$$

where $1_{[k \neq a]} \in \{0, 1\}$ is an index function that equals 1 only if $k \neq a$. The final Loss function is calculated on all positive samples and is called Multiple Negatives Ranking Loss (Song, Hong and Jung, 2024).

3.3. Dual Channel Knowledge Augmentation Module

This paper constructed a dual channel knowledge augmentation module, which includes a dual channel network and a shared attention layer. The module structure is illustrated in Figure 3 as "Dual Channel Knowledge Augmentation." In Figure 4, the dual-channel knowledge augmentation module and the contrastive learning module work together to extract the deep toxicity of embeddings and contribute to classifying hard samples. The toxic text and non-toxic text are embedded into vectors of A and B, which are sample A and sample B. Then, the similarity of vector A is distanced

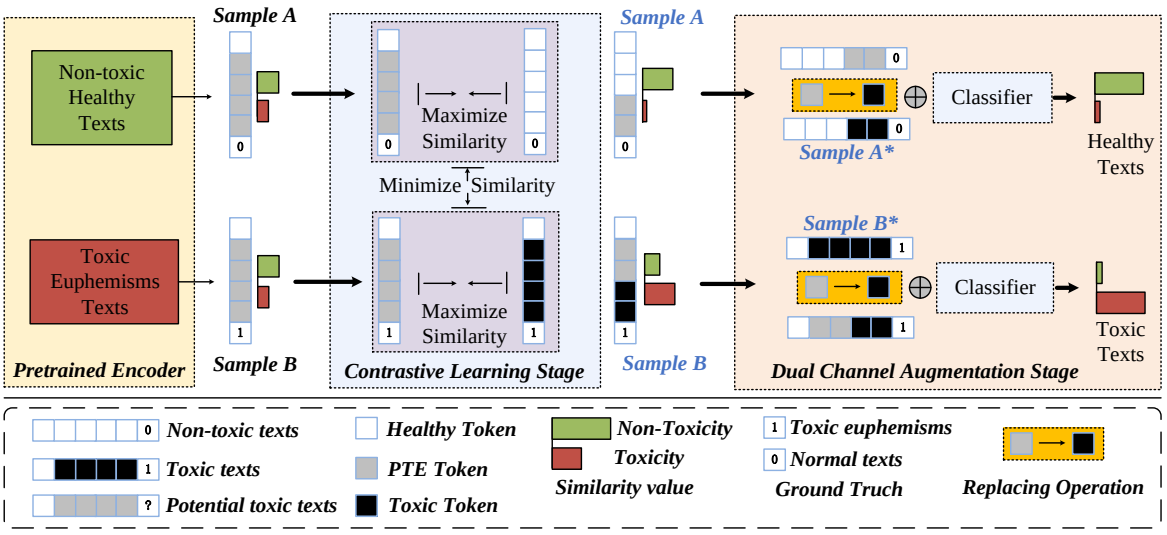


Figure 4: Data flow diagram of comments in the TED-SCL framework. Sample A and Sample B are comments containing potential toxic euphemism terms. Enhanced by the contrastive learning module, the vectors of both samples are pushed apart and aligned with their true semantic embeddings. The dual-channel knowledge augmentation module uses toxic euphemism annotations to generate a comparative sample for each text. Together, these two modules enable the classifier to more effectively capture the underlying toxic information in the target comments.

from toxic ones and closer to non-toxic ones, while B is closer to toxic ones and distanced from non-toxic ones after the contrastive learning stage. After that, the dual-channel network constructs another comparable sample* for A and B by replacing the PTET with annotations. These two modules work together to make the classifier more effective at capturing the underlying toxic information in target comments.

3.3.1. Dual Channel Augmentation Layer

The dual channel augmentation layer includes shallow semantic channel and deep semantic channel. The shallow channel extracts healthy semantic vectors of the original toxic euphemism comments without euphemisms replaced, and the deep semantic channel extracts toxic semantic vectors of the replaced toxic euphemisms comments with prompted euphemistic paragraphs. The background knowledge of potential toxic euphemisms in PTETD is the explanation of PTETs in comments, as shown in Equation 10. The dual channel knowledge augmentation module combines two LSTM network with a shared parameters attention layer (Liu, Deng, Cheng, Ren, Wang and Zhang, 2024). We adopted the dual channel network to integrate background knowledge on toxic euphemisms. This approach excels in generalization and robustness because the dual channel network maintains a better balance between fusion performance and parameter size.

Specifically, the dual channel takes different version of the same toxic euphemisms comment and feeds them into the fusion network. Leveraging its memory function, the dual channel network further learns the shallow features and deep features of toxic euphemisms comments, and then obtains vector pairs with toxic euphemisms knowledge. $H = \{h_1, h_2, h_3, ..., h_k\}$, where k represents the number of hidden nodes in the network, and the calculation method for h_i and h'_i is as follows:

$$h_i = \sigma(W_i, h_{i-1}) \tag{8}$$

$$h'_i = \sigma(W'_i, h'_{i-1}) \tag{9}$$

$$W'_i = \sum_{i=1}^{k} (W_i, W_{toxic}) \tag{10}$$

The semantic vector pairs V_{LSTM} and $V_{LSTM'}$ are obtained from the dual channel network and contacted to obtain the fused feature vector V_{joint}, shown as follows:

$$V_{LSTM} = \{h_1, h_2, h_3, ..., h_k\} \tag{11}$$

$$V_{LSTM'} = \{h'_1, h'_2, h'_3, ..., h'_k\} \tag{12}$$

$$V_{\text{joint}} = V_{LSTM} \oplus V_{LSTM'}. \tag{13}$$

3.3.2. Shared Attention Mechanism

The fused toxic euphemisms vector was input into the shared attention mechanism layer. Based on the sentiment analysis of toxic euphemisms, the sentiment of toxic euphemisms comments is contacted with the toxic euphemisms vector at the same time. Given that different semantic vectors including the complex toxic euphemisms information (Choi, Kim, Han, Xu and Lee, 2020), the shared attention mechanism assigns adaptive weights to the significant features of toxic euphemisms vector pairs. The attention layer shares parameters with the dual channel network, which distinguishes and extracts subtle toxic euphemism vector pairs from the dual channel. as shown below:

$$u_i = \tanh(w_w \bullet F_i + b_w) \tag{14}$$

where w_w and b_w represent the weight matrix and bias term, and u_i is the middle hidden layer vector of F_i,

$$\alpha_i = \frac{\exp({u_i}^T, u_w)}{\sum_t \exp({u_i}^T, u_w)} \tag{15}$$

where u_w is a randomly initialized vector that has been optimized as a model parameter during training. The weights α_i are implemented by the Softmax function, which can determine the weight α_i of the output vector F_i by calculating the similarity of the intermediate vectors u_i and u_w,

$$F_E = \sum_t \alpha_i \bullet F_i \tag{16}$$

where F_E is the vector for the final judgment of whether the comment is a malicious euphemism; t represents the top layer; $\tanh(\cdot)$ is the scaling factor and the hyperbolic tangent function.

3.3.3. Multi-layer Perceptron

Finally, the dimension of the output vector of the attention mechanism layer is reduced through the multi-layer perceptron, and the feature vector is input into the Sigmoid function to obtain the probability of toxic euphemism, shown as follows:

$$\overline{F}_E = MLP(F_E \cdot W + b) \tag{17}$$

$$p_d = \text{Sigmoid}(\bar{F}_E) \tag{18}$$

where $\bar{F}_E$ is the vector that ultimately determines whether the comment is a toxic euphemism, and p_d is the probability that the comment is a toxic euphemism.

The optimization objective of the model is to minimize the cross-entropy loss function, shown as follows:

$$L = -\sum_{d \in D} [y_d \log p_d + (1 - y_d) \log(1 - p_d)] \tag{19}$$

where D is the sample data set, d is the sample, $\bar{y}_d$ is the true value of the sample, $\bar{p}_d$ is the probability that the sample d is predicted to be positive class, and L is the objective to minimize the cross-entropy loss.

Table 4
Partitioning of the training dataset, validation dataset and test dataset in TE-Dataset. The table shows the number of positive, negative, and the total number of samples in three subsets.

Dataset Partitioning	Samples Proportion		
	Tox.	N-Tox.	Total
Training dataset	6,340	5,042	11,382
Validation dataset	2,113	1,681	3,794
Test dataset	2,112	1,683	3,795
Ratio	55.71%	44.29%	100%

4. Experiments

4.1. Experimental Setup

Our models were built using the PyTorch framework and the Scikit-learn library. All experiments were conducted on a server equipped with an Intel(R) Xeon(R) E5410 CPU, three NVIDIA Tesla V100 GPUs with 32GB of memory each, and 128GB of system memory.

Experimental Dataset. The TE-Dataset details are provided in Table 4, which was randomly split into training, validation, and test sets in a ratio of 6:2:2. **Baseline Setup.** To demonstrate the superior performance of the proposed TED-SCL framework, we conducted a comparative analysis with several baselines. These methods include traditional static embedding, newer pretrained models and currently advanced LLMs. An overview of each baseline is provided below:

- Perspective-API (Clarke, Hall, Mittal and et al., 2023): A toxic language detection API launched by Google.
- Tencent-API (Wang et al., 2023): A widely used API for toxic content analysis developed by Tencent.
- Machine Learning (ML): There are classic ML methods which are effective and efficient: k Nearest Neighbor (k-NN) (Sumanth, Samiuddin and Jamal, 2022), AdaBoost (Wyner, Olson, Bleich and Mease, 2017), Random Forest (RF) (Chaudhary, Kolhe and Kamal, 2016), Support Vector Machine (SVM) (Gaydhani, Doma, Kendre et al., 2018)
- TextFNN (Pérez and Luque, 2019): A method based on dense layers and a dropout layer.
- TextCNN (Li, Du and Ji, 2020a): A method based on convolutional layers and a dropout layer.
- LSTM (Dubey, Nair and Khan, 2020): A type of recurrent neural network (RNN) architecture designed to capture long-range dependencies in sequential data.
- BiLSTM (Dessì, Recupero and Sack, 2021): Bidirectional Long Short-Term Memory (BiLSTM) by processing sequences in both forward and backward directions.
- LSTM-Att (Dai, Tao, Yan, Feng and Chen, 2023): An improved method combined LSTM with attention mechanism.
- BiLSTM-Att (Neog and Baruah, 2024): An improved method combined BiLSTM with attention mechanism.
- BERT-base (Devlin et al., 2019): The pre-trained language model based on the Transformer architecture, utilizing bidirectional masked language modeling.
- BERT-large (Devlin et al., 2019): The extended version of BERT with more parameters and a deeper network, allowing it to capture more complex language features.
- RoBERTa-base (Cui et al., 2020): The improved version of BERT that removes the next-sentence prediction task and trains on more data for better performance.

Table 5
Details of pretrained models in experiments. "Dimension" represents the word embedding dimension of each model. "Level" is the word segmentation level by tokenizer of each model.

Pretrained Models	Configuration			
	Architecture	Parameters	Embedding Dimension	Language
bert-base-chinese	BERT	110M	768	Chinese
bert-large-chinese	BERT	178M	1024	Chinese
chinese-roberta-wwm-ext-base	RoBERTa	125M	768	Chinese
chinese-roberta-wwm-ext-large	RoBERTa	355M	1024	Chinese
sbert-chinese-general-v2	Sentence-BERT	110M	768	Chinese
bert-base-cased	BERT	110M	768	English
bert-large-cased	BERT	340M	1024	English
xlm-roberta-base	RoBERTa	270M	768	Multilingual
xlm-roberta-large	RoBERTa	550M	1024	Multilingual

- RoBERTa-large (Cui et al., 2020): The extended version of RoBERTa with more parameters and a deeper structure, further improving its language understanding capabilities.
- SBERT (Sentence-BERT) (Madhu, Satapara and Modha, 2023): SBERT uses Siamese and Triplet Networks to generate semantically meaningful sentence embeddings.
- GPT-4o (OpenAI, 2023): Currently advanced LLMs developed by OpenAI, more accurate and versatile in NLP tasks compared to earlier versions.
- GPT-4-Turbo (OpenAI, 2024): A variant of GPT-4, optimized for performance and cost-efficiency.
- QWen2-7B (Yang, Yang and Hui, 2024): An open source LLM developed by Tongyi lab, performs well in Chinese tasks.
- QWen2-72B (Yang et al., 2024): Another version of Qwen2 model, more parameters and higher performance.
- Llama3-8B (Touvron, Lavril and et al., 2023): An advanced open source LLM developed by Meta, providing strong text comprehension and generation capabilities.
- Llama3-70B (Touvron et al., 2023): Another version of Llama3 model, better and bigger.
- ChatGLM3-6B (Du, Qian and Liu, 2022): An advanced open source LLM developed by Tsinghua university and ZHIPU AI.

Since the prompt templates used in the large language model influence the results of toxic euphemism detection, we compared several templates and selected the most effective ones for experiments on both English and Chinese datasets. In the template, the "sentence" refers to the input comment, "PTET" represents the toxic euphemism term within the comment, and the "annotation" refers to the toxic information annotated for the PTET in the comment. Additionally, to ensure fairness in the large language model experiments, we used the open-source large model API platform , which provided access to nearly all of the latest deployed LLMs.

Evaluation Metrics. The training parameters of our proposed method in experiments are shown in Table 6. This paper employs seven metrics to evaluate the performance of the detection model: *Accuracy* (ACC.), *Precision* (Pre.), *Recall* (Rec.), *F1 score* (F1.), *N-Precision* (N-Pre.), *N-Recall* (N-Rec.), and *N-F1 score* (N-F1.). The calculation formulas are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

Explanation: Toxic euphemism language is implicit and toxic, which usually attacks others with toxic euphemism terms.
Task: Please judge the sentence: {Sentence.}, if the above sentence is toxic euphemism, you can only output “1”. If not , you can only output “0”.

a) The non-prompted template used in our experiments of English Datasets.

Explanation: Toxic euphemism language is implicit and toxic, which usually attacks others with toxic euphemism terms. The potential toxic euphemism of “{PTET}” is “{Annotation}”.
Task: Please judge the sentence: {Sentence}, if the above sentence is toxic euphemism, you can only output “1”. If not , you can only output “0”.

b) The prompted template used in our experiments of English Datasets.

解释：有害委婉语是隐性的和有害的，通常利用其中的有害委婉术语来攻击他人。
任务：请判断句子：{Sentence}，如果上面的句子是有害委婉语，你只能输出“1”，否则只能输出“0”。

c) The non-prompted template used in our experiments of Chinese Datasets.

解释：有害委婉语是隐性的和有害的，通常利用其中的有害委婉术语来攻击他人。其中，潜在术语“{PTET}”的含义是“{Annotation}”。
任务：请判断句子：{Sentence}，如果上面的句子是有害委婉语，你只能输出“1”，否则只能输出“0”。

d) The prompted template used in our experiments of Chinese Datasets.

Figure 5: The LLM templates used in the experiments included both non-prompted and prompted versions. While both templates explained toxic euphemism language, the prompted template provided additional annotations for PTET (toxic euphemism term) within the sentence. The "sentence" refers to a comment, "PTET" represents the toxic euphemism term in the comment, and the "annotation" refers to the detailed information provided for the PTET in the sentence.

Table 6
Details of training configuration. The "Fine-tuning Stage" is the training stage of contrastive learning, and the "Training Stage" is the training stage of classifier.

Fine-tuning Stage		**Training Stage**	
Parameters	Default	Parameters	Default
Batch size	16	Batch size	256
Learning rate	2e-5	Learning rate	1e-3
Epochs	10	Epochs	200
Loss function	InfoNCE	Loss function	BCELoss
Max sequence	128	Max sequence	128
Optimization	Adam	Optimization	Adam
Cutoff rate	0.15	Dropout rate	0.5

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F1score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{23}$$

$$N - Precision = \frac{TN}{TN + FN} \tag{24}$$

$$N - Recall = \frac{TN}{TN + FP} \tag{25}$$

$$N - F1score = \frac{2 \times N - Precision \times N - Recall}{N - Precision + N - Recall} \tag{26}$$

where *TP* (True Positive) is the number of comments predicted as toxic euphemisms, *FN* (False Negative) is the number of comments predicted as non-toxic euphemisms, *FP* (False Positive) is the number of non-toxic euphemisms predicted as toxic euphemisms, and *TN* (True Negative) is the number of non-toxic euphemisms predicted as non-toxic euphemisms.

For fairness in the experiments, the same maximum number of training epochs, learning rate, and input batch size were used across all experiments. Each method was independently tested 10 times in specific experiments, and the final results were averaged over these 10 times to ensure consistency and reliability.

4.2. Results on Detection Performance

To demonstrate the toxic euphemism detection performance of TED-SCL, we compared our proposed method to comprehensive baselines, especially advanced large language models (LLMs). The LLMs includes GPT-4-o[3], Qwen2-7B[4], GPT-4-Turbo[5], Qwen2-72B[6], Llama3-8B[7], Llama3-70B[8], ChatGLM3-6B[9]. Table 5 provides details of the specific pretrained models employed in our experiments.

In Table 7, The results show that the classifier performance of TED-SCL outperforms other baselines including all advanced LLMs on majority of metrics. The second best method is RoBERTa-large (Cui et al., 2020). Additionally, we found that models with larger scale of parameters perform better: TED-SCL-large outperforms TED-SCL-base, and RoBERTa-large outperforms RoBERTa-base, and it is similar to LLMs: Qwen2-7B outperforms Qwen2-72B, Llama3-8B outperforms Llama3-70B. Tencent-API (Wang et al., 2023) and Perspective-API (Clarke et al., 2023) are ranked as the worst and second-worst performers. This might be attributed to the fact that their own training datasets scarcely contain toxic euphemistic language.

Surprisingly, the performance of SBERT (Madhu et al., 2023) is inferior to traditional detection methods. We think that SBERT's direct sentence-level averaging further weakens the discriminative features between different comments, leading to poorer performance compared to BERT and RoBERTa. The result directly demonstrates the TED-SCL's capacity to extract toxic euphemism comments embeddings that outperforms traditional average pooling. With TED-SCL-large achieving over 93% on four metrics and over 94% on three metrics, the experimental results significantly demonstrate TED-SCL's superior performance for detecting toxic euphemism.

4.3. Generalization Analysis

Due to the varying characteristics of toxic euphemisms across different topics, it is essential for a detection method to effectively adapt to zero-shot samples. To demonstrate the effectiveness of our proposed detection method in distinguishing toxic euphemisms across topics, we conducted cross-topic transfer experiments using datasets from three different topics. Specifically, we train our models on datasets from one topic and evaluate their performance on datasets from other topics. The experiment utilizes datasets from three topics: race, gender, and general. The evaluation metrics are Accuracy, Recall, and F1 score. Tencent-API (Wang et al., 2023) and Perspective-API (Clarke et al., 2023) performed the worst performance in the above comparison experiment, so we removed them from the baseline in other experiments. The results demonstrate that our proposed detection method maintains high detection accuracy and transferability across various topics, outperforming other classic baselines. Additionally, our method surpasses Llama3 and ChatGLM3, and performs comparably to advanced LLMs, such as GPT-3.5 Turbo and Qwen2-7b.

in Table 8 and 9, TED-SCL consistently achieves the highest detection accuracy when transferring between three topics. Specifically, it outperforms other traditional methods by at least 5% in precision across all zero-shot scenarios.

[3] https://openai.com/index/hello-gpt-4o/
[4] https://huggingface.co/Qwen/Qwen2-7B-Instruct/
[5] https://platform.openai.com/docs/api-reference/introduction
[6] https://huggingface.co/Qwen/Qwen2-72B-Instruct/
[7] https://huggingface.co/Groq/Llama-3-Groq-8B-Tool-Use/
[8] https://huggingface.co/Groq/Llama-3-Groq-70B-Tool-Use/
[9] https://huggingface.co/THUDM/chatglm3-6b/

Table 7
The results of experiment on model detection performance. The baselines are classified six categories. "Features" represents dataset features used as inputs, "Tex." represents text, "Sen." represents sentimental signal, and "Eup." represents toxic euphemism annotation signal. Other columns are evaluation metrics.

Category	Methods	Features			ACC.(%)	Tox.			N-Tox.		
		Tex.	Sen.	Eup.		Pre.	Rec.	F1.	N-Pre.	N-Rec.	N-F1.
API	Perspective-API (Clarke et al., 2023)	✓	✓		61.33	85.89	16.75	28.04	58.95	**97.75**	73.55
	Tencent-API (Wang et al., 2023)	✓	✓		62.64	66.67	33.84	44.89	61.42	86.16	71.72
Machine Learning	RF (Chaudhary et al., 2016)	✓			81.42	79.57	78.97	79.27	82.90	83.42	83.16
	SVM (Gaydhani et al., 2018)	✓			69.98	66.63	66.67	66.65	72.72	72.69	72.71
	k-NN (Sumanth et al., 2022)	✓			75.49	69.65	80.67	74.76	81.84	71.25	76.18
	AdaBoost (Wyner et al., 2017)	✓			67.58	64.15	63.33	63.74	70.32	71.06	70.69
Toxicity Classifier	TextFNN (Pérez and Luque, 2019)	✓			84.01	84.07	81.04	82.53	83.91	86.56	85.21
	TextCNN (Li et al., 2020a)	✓			73.10	71.23	67.39	69.25	74.45	77.73	76.05
	LSTM (Dubey et al., 2020)	✓			87.80	87.65	85.54	86.58	87.88	89.69	88.78
	LSTM-Att (Dai et al., 2023)	✓			87.27	83.78	87.36	85.53	90.09	87.16	88.60
	BiLSTM (Dessì et al., 2021)	✓			87.96	87.88	85.67	86.76	87.98	89.87	88.92
	BiLSTM-Att (Neog and Baruah, 2024)	✓			88.19	88.35	85.79	87.05	88.03	90.23	89.12
	BERT-base (Devlin et al., 2019)	✓			88.81	86.69	88.99	87.60	90.68	88.69	89.50
	BERT-large (Devlin et al., 2019)	✓			88.37	81.36	92.06	86.14	94.32	85.90	89.77
	RoBERTa-base (Cui et al., 2020)	✓			89.97	87.56	90.61	88.90	92.05	89.39	90.59
	RoBERTa-large (Cui et al., 2020)	✓			89.01	85.66	90.67	87.95	91.90	87.53	89.55
	SBERT (Madhu et al., 2023)	✓			86.53	84.43	85.43	84.92	88.22	87.38	87.80
Large Language Model	GPT-4o (OpenAI, 2023)	✓			71.20	62.78	88.29	73.38	85.65	57.18	68.58
	GPT-4-Turbo (OpenAI, 2024)	✓			76.39	75.30	70.67	72.91	77.15	81.03	79.05
	QWen2-7B (Yang et al., 2024)	✓			64.08	57.62	76.17	65.61	73.54	54.17	62.38
	QWen2-72B (Yang et al., 2024)	✓			67.75	59.96	85.13	70.36	81.47	53.50	64.59
	Llama3-8B (Touvron et al., 2023)	✓			54.78	48.26	7.32	12.71	55.24	93.58	69.48
	Llama3-70B (Touvron et al., 2023)	✓			54.84	48.29	5.80	10.35	55.19	94.92	69.80
	ChatGLM3-6B (Du et al., 2022)	✓			57.42	67.30	10.36	17.96	56.67	95.88	71.23
Prompted Large Language Model	GPT-4o (OpenAI, 2023)	✓		✓	72.54	63.06	**93.97**	75.48	91.77	54.98	68.76
	GPT-4-Turbo (OpenAI, 2024)	✓		✓	77.26	70.67	84.48	76.96	84.89	71.31	77.51
	QWen2-7B (Yang et al., 2024)	✓		✓	66.77	61.35	70.55	65.63	72.54	63.65	67.81
	QWen2-72B (Yang et al., 2024)	✓		✓	70.17	61.86	87.82	72.59	84.83	55.70	67.24
	Llama3-8B (Touvron et al., 2023)	✓		✓	54.97	49.40	4.86	8.85	55.21	95.53	70.08
	Llama3-70B (Touvron et al., 2023)	✓		✓	55.13	51.45	4.16	7.69	55.29	96.79	70.34
	ChatGLM3-6B (Du et al., 2022)	✓		✓	56.52	64.32	7.49	13.42	56.07	96.60	70.96
Proposed	**TED-SCL-base (Ours)**	✓	✓	✓	92.36	90.04	92.40	91.39	93.92	92.28	93.09
	TED-SCL-large (Ours)	✓	✓	✓	**93.94**	**93.09**	93.36	**93.23**	**94.59**	94.36	**94.48**

Notably, significant performance disparities are observed when our detection method is applied across different topics. Notably, significant performance differences are observed when applying detection methods across different topics. For example, RoBERTa-large-avg (Cui et al., 2020) achieves higher precision when transitioning from race to gender but suffers a notable 15% drop in precision when transferring from general to gender or race. These results indicate that model generalization is influenced by variations in topic categories. In summary, the findings confirm the superior and stable generalization of our method, underscoring its effectiveness in detecting toxic euphemisms across diverse topic.

4.4. Robustness Evaluation

Supervised methods for detecting toxic euphemisms are highly sensitive to the annotation noise, categories balance and samples scale of datasets. To demonstrate the robustness of TED-SCL, we conducted three different experiments about annotation noise, categories balance, and samples scale.

Table 8
The results of experiment on model generalization performance. $D_1 \rightarrow D_2$ is training the model on dataset D_1 and testing it on dataset D_2. Acc., Pre., and F1. represent accuracy, precision and F1 score. $\overline{Acc.}$ is the weighted macro average accuracy.

Methods	*Race→Gender*			*Race→General*			*Gender→Race*			$\overline{Acc.}$
	Acc.	Pre.	F1.	Acc.	Pre.	F1.	Acc.	Pre.	F1.	
TextFNN (Pérez and Luque, 2019)	64.07	61.41	56.90	58.62	56.73	49.50	63.19	61.23	62.19	62.12
TextCNN (Li et al., 2020a)	60.50	62.35	58.05	59.75	60.25	60.05	62.45	63.10	61.80	53.26
LSTM (Dubey et al., 2020)	65.49	62.10	60.27	56.69	53.97	51.47	64.59	61.31	63.53	62.78
BiLSTM (Dessì et al., 2021)	65.78	61.47	62.06	56.12	52.99	54.49	64.53	60.85	64.60	62.96
LSTM-Att (Dai et al., 2023)	63.14	61.16	55.30	57.01	55.04	53.25	62.69	61.92	56.70	62.47
BiLSTM-Att (Neog and Baruah, 2024)	65.47	61.71	62.06	54.11	51.83	54.49	64.28	61.76	61.86	62.71
BERT-base (Devlin et al., 2019)	70.44	66.40	65.24	62.25	59.48	55.76	66.51	64.67	61.95	65.89
BERT-large (Devlin et al., 2019)	68.57	65.28	62.71	60.15	57.82	56.69	65.48	63.32	62.04	64.87
RoBERTa-base (Cui et al., 2020)	70.15	66.63	64.29	67.16	62.69	63.48	67.18	64.47	64.14	66.09
RoBERTa-large (Cui et al., 2020)	71.69	68.68	65.32	68.37	62.78	65.60	67.29	66.03	62.06	66.88
SBERT (Madhu et al., 2023)	64.45	62.26	56.50	56.85	55.44	47.42	63.49	62.07	59.20	62.43
GPT-4o (OpenAI, 2023)	72.00	65.44	77.02	84.05	76.62	86.13	72.39	66.62	78.36	74.24
QWen2-7B (Yang et al., 2024)	63.05	62.50	64.22	67.24	67.20	68.57	66.14	64.65	70.92	58.74
QWen2-72B (Yang et al., 2024)	68.95	63.99	73.80	80.17	73.55	82.91	72.10	67.36	77.40	73.16
Llama3-8B (Touvron et al., 2023)	48.95	44.12	10.03	46.98	30.00	4.62	47.37	52.10	10.61	48.95
Llama3-70B (Touvron et al., 2023)	49.90	51.43	12.00	47.84	41.67	7.58	46.67	34.78	10.89	49.51
ChatGLM3-6B (Du et al., 2022)	51.62	63.16	15.84	46.55	14.29	1.57	51.74	74.44	21.58	50.73
TED-SCL (Ours)	69.43	67.37	61.35	70.87	65.87	66.22	67.70	65.53	63.68	70.04

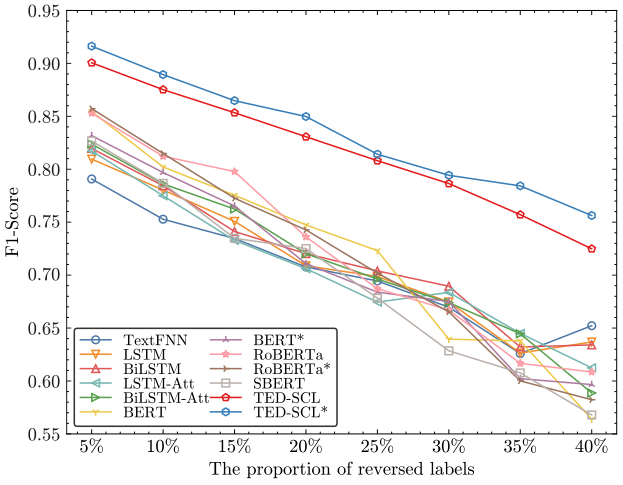


Figure 6: Results of the experiment on the proportion of reversed labels. The vertical axis of the chart represents the F1 score, while the horizontal axis shows different ratios of correctly labeled samples that are reversed to incorrect labels. Each method was executed 10 times under identical parameter settings, with the final result being the average of these 10 runs. The experiment demonstrates that TED-SCL continues to perform well even when label confusion occurs in the training samples.

4.4.1. Proportion of Reversed Labels

We simulate the noise of dataset annotations in real detection scenario by changing ground truth of dataset samples. The specific way is to change the label of the positive sample from "1" to "0", and the label of the negative sample from "0" to "1", which is called ”reversed“. In the experiment on different proportion of reversed labels, the F1 score

Table 9
The results of experiment on model generalization performance. $D_1 \rightarrow D_2$ denotes training the model on dataset D_1 and testing it on dataset D_2. Acc., Pre., and F1. represent accuracy, precision and F1 score. $\overline{Acc.}$ is the weighted macro average accuracy.

Methods	*Gender→General*			*General→Race*			*General→Gender*			$\overline{Acc.}$
	Acc.	Pre.	F1.	Acc.	Pre.	F1.	Acc.	Pre.	F1.	
TextFNN (Pérez and Luque, 2019)	56.85	53.91	52.50	55.87	54.26	55.27	55.28	55.20	52.08	50.20
TextCNN (Li et al., 2020a)	55.14	52.37	50.88	54.22	52.61	53.56	53.57	53.46	50.56	48.61
LSTM (Dubey et al., 2020)	56.85	54.52	49.78	55.60	53.85	53.93	55.03	51.73	49.98	50.00
BiLSTM (Dessì et al., 2021)	56.93	54.05	56.02	55.84	54.63	54.22	54.46	51.10	47.13	50.00
LSTM-Att (Dai et al., 2023)	57.41	57.30	43.85	55.86	54.60	53.28	55.26	51.76	48.83	50.20
BiLSTM-Att (Neog and Baruah, 2024)	55.76	53.57	53.59	56.24	54.40	55.03	53.97	50.92	48.85	49.90
BERT-base (Devlin et al., 2019)	61.85	58.61	59.06	56.87	55.67	54.30	60.06	55.80	57.68	52.70
BERT-large (Devlin et al., 2019)	62.89	60.18	57.43	55.97	54.43	55.97	57.21	53.45	53.64	51.40
RoBERTa-base (Cui et al., 2020)	60.27	59.29	52.89	56.64	55.10	53.30	57.11	53.90	54.91	51.50
RoBERTa-large (Cui et al., 2020)	64.63	62.64	59.27	56.24	55.27	51.99	58.43	54.97	55.10	52.20
SBERT (Madhu et al., 2023)	50.73	48.37	40.72	53.66	52.39	54.51	53.76	51.06	54.75	47.90
GPT-4o (OpenAI, 2023)	84.05	76.62	86.13	72.39	66.62	78.36	72.00	65.44	77.02	66.20
QWen2-7B (Yang et al., 2024)	67.24	67.20	68.57	60.14	64.65	70.92	63.05	62.50	64.22	55.40
QWen2-72B (Yang et al., 2024)	80.17	73.55	82.91	72.10	67.36	77.40	68.95	63.99	73.80	64.80
Llama3-8B (Touvron et al., 2023)	46.98	30.00	4.62	47.37	52.10	10.61	48.95	44.12	10.03	42.90
Llama3-70B (Touvron et al., 2023)	47.84	41.67	7.58	46.67	34.78	10.89	49.90	51.43	12.00	42.90
ChatGLM3-6B (Du et al., 2022)	46.55	14.29	1.57	51.74	74.44	21.58	51.62	63.16	15.84	45.90
TED-SCL (Ours)	71.96	68.24	67.02	59.45	59.07	53.98	64.05	61.16	58.19	56.10

is the evaluation metric. The proportion of reversed labels in the training data is 5%, 10%, 15%, 20%, 25%, 30%, 35%, and 40%. The results of experiment are in Figure 6.

The results show that as the proportion of noisy labels increases, the F1 score of all detection methods decline. However, TED-SCL demonstrates greater stability compared to the baseline methods. Specifically, both TED-SCL-base and TED-SCL-large consistently achieve higher F1 score across different noise ratios than all baseline methods. These findings highlight the effectiveness of our proposed detection method in reducing the impact of label noise when identifying toxic euphemisms."

4.4.2. Proportion of Positive and Negative Samples

Due to the significant imbalance in the distribution of toxic euphemisms and normal comments on real social platforms, toxic euphemisms are typically less frequent than non-euphemisms. To assess how this imbalance affects the performance of detection methods and to demonstrate the effectiveness of our proposed detection method under varying imbalance ratios in real-world scenarios, we randomly partitioned toxic euphemism comments and non-toxic comments in the training set of the TE-Dataset into imbalanced training sets at different ratios. For instance, we set a 1:1 ratio, comprising 5,042 toxic euphemism comments and an equal number of non-toxic comments. In contrast, a 5:1 ratio includes 5,042 toxic euphemism comments and 1,008 non-toxic comments. Detailed results under these different ratios of toxic euphemisms to non-toxic comments are presented in Figure 7.

The results of this experiment indicate that the more unbalanced the distribution of toxic euphemism comments and non-toxic comments, the greater the decrease in the performance of the detection method. However, the TED-SCL method demonstrates better stability under different imbalance ratios in the TE-Dataset. It exhibits significantly higher F1 score in different ratio of toxic euphemism comments and non-toxic comments, ranging from 5:1 to 1:5. It is worth noting that, when there are more non-toxic comments than toxic euphemisms, the baseline methods are more affected by the imbalance. In general, the proposed detection method effectively adapts to the challenge of unbalanced distribution of toxic euphemism comments, maintaining stable detection performance, which further confirms its high generalization ability.

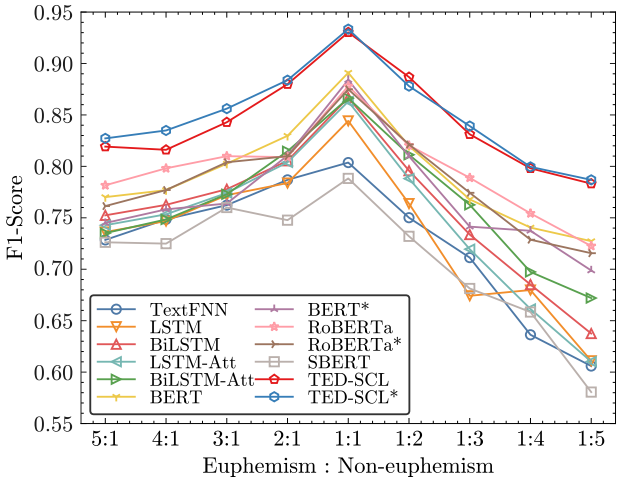


Figure 7: Results of the experiment with different proportions of positive and negative training samples. The vertical axis of the chart represents the F1 score, and the horizontal axis shows different proportions of positive and negative samples. Each method was executed 10 times with identical parameter settings, and the final result was obtained by averaging the outcomes of these 10 runs. The experiment demonstrates that TED-SCL performs well even under conditions of an unbalanced dataset.

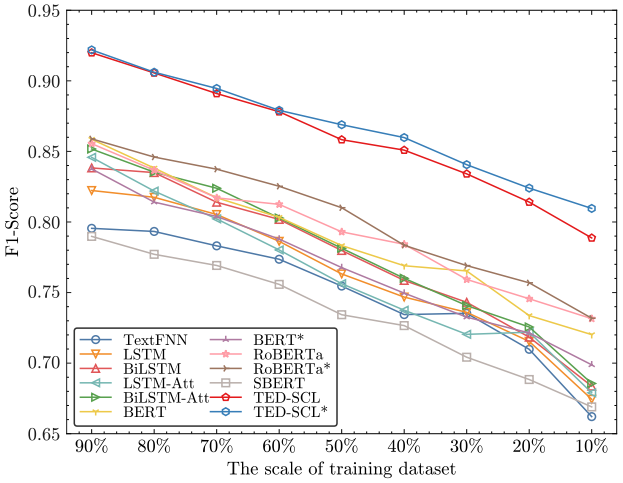


Figure 8: Results of experimental on different scale of training dataset. The vertical coordinate of the chart is F1 score, and the horizontal coordinate represents different removal proportion of TE-Dataset. Each method is executed 10 times under identical parameter settings, and the final result is obtained by averaging the outcomes of these 10 runs. The experiment shows that TED-SCL still performs well under the condition of few-shot.

4.4.3. Scale of Training Dataset

Given the limited scale of annotated toxic euphemism datasets on social networks and the reliance of existing methods on large datasets, it is important to evaluate how detection performance varies with dataset size. Additionally, it is necessary to show that the proposed method can achieve high performance even with limited data. To examine the impact of dataset scale on detection methods and highlight the effectiveness of our approach, we gradually reduce the training data from the TE-Dataset in 10% increments and evaluate the performance of different methods at each scale. The experimental results are presented in Figure 8.

From the results of this experiment, it is evident that the TED-SCL-base outperforms other baselines in different removed proportions of the training set. Specifically, under the conditions of removing data in proportions ranging from 10% to 90%, both TED-SCL-base and TED-SCL-large significantly outperform other detection methods. In general,

Table 10
The results of experiment comparing detection performance of methods on English euphemisms datasets. "EACL", "FigLang", and "JointEDI" are three different English datasets from social media platforms.

Methods	EACL			FigLang			JointEDI		
	Pre.	Rec.	F1.	Pre.	Rec.	F1.	Pre.	Rec.	F1.
TextFNN (Pérez and Luque, 2019)	80.86	83.42	82.12	72.58	72.87	72.71	86.40	86.51	86.45
TextCNN (Li et al., 2020a)	80.85	85.31	82.99	71.95	77.73	74.69	86.74	89.90	88.28
LSTM (Dubey et al., 2020)	74.68	**100.00**	85.51	60.97	65.22	61.78	67.26	69.34	67.17
LSTM-Att (Dai et al., 2023)	81.30	82.05	81.64	73.89	71.78	72.74	85.54	87.17	86.31
BiLSTM (Dessì et al., 2021)	80.33	80.45	80.21	65.04	64.75	64.86	83.23	85.26	84.21
BiLSTM-Att (Neog and Baruah, 2024)	80.11	82.88	81.46	74.68	73.63	74.08	86.77	85.61	86.18
BERT-base (Devlin et al., 2019)	83.91	88.15	85.93	72.71	81.83	76.86	86.48	**90.87**	88.60
BERT-large (Devlin et al., 2019)	84.12	86.75	85.39	**78.69**	80.38	79.38	87.54	87.78	87.63
RoBERTa-base (Cui et al., 2020)	81.45	90.99	85.85	70.14	87.08	77.58	85.82	83.69	84.71
RoBERTa-large (Cui et al., 2020)	82.18	93.56	87.47	72.48	82.40	76.80	85.11	88.44	86.66
GPT-4o (OpenAI, 2023)	84.09	88.70	86.33	72.63	92.08	**81.20**	86.83	58.38	69.82
GPT-4-Turbo (OpenAI, 2024)	83.40	75.68	79.35	73.05	87.43	79.60	92.00	43.14	58.89
QWen2-7B (Yang et al., 2024)	80.15	74.66	77.30	67.46	92.89	78.16	83.93	61.96	71.29
QWen2-72B (Yang et al., 2024)	83.75	79.45	81.55	68.43	**95.35**	79.68	79.95	58.57	67.61
Llama3-8B (Touvron et al., 2023)	75.40	94.52	83.89	61.66	72.95	66.83	**96.21**	77.81	86.04
Llama3-70B (Touvron et al., 2023)	77.42	73.97	75.66	66.88	82.24	73.77	79.92	50.76	62.09
ChatGLM3-6B (Du et al., 2022)	71.56	50.0	58.87	67.34	72.13	69.66	90.79	68.84	78.31
TED-SCL (Ours)	**85.16**	90.41	**87.71**	77.47	83.61	80.42	89.48	88.79	**89.13**

as the scale of the removed data increases, almost all methods experience a decrease in detection performance. In conclusion, the proposed detection methods effectively utilize training data on a limited scale, demonstrating better detection performance.

4.4.4. Performance of methods on different datasets

Because toxic euphemisms exhibit varying hidden characteristics across different language English, it is necessary to evaluate the performance of TED-SCL in detecting toxic euphemisms in other language datasets.. To evaluate the effectiveness and adaption of the TED-SCL to different language datasets, we conducted the experiment comparing performance of methods on different datasets. Due to the lack of English toxic euphemism datasets, we chose three new English euphemism datasets: EACL (Lee, Trujillo and Plancarte, 2024), FigLang (He, Vieira and Garcia, 2024), and JointEDI (Hu, Li and Wang, 2024a). These datasets were collected from different English language forums and social media platforms, including blogs (EACL), articles from the social news platform Reddit (FigLang), the dark website (JointEDI), and hand-collected euphemistic expressions (JointEDI). Although these English datasets obtained from social media platforms almost have no toxic content, they have four feature columns including texts, candidates, target, and euphemism label. So they can be used to evaluate the model's ability of extracting euphemism information. In this experiment, we used the pretrained model of xlm-roberta-base as our base encoder.

As shown in Table 10, our proposed method exceeds both traditional baselines and advanced LLMs on most metrics. Firstly, TED-SCL achieved SOTA performance for F1 score on both EACL and JointEDI datasets. The results show that TED-SCL has wider applicability to cross-language datasets due to the use of the contrastive learning module and dual-channel knowledge augmentation module. Secondly, TED-SCL achieved the second-best Precision and F1 score on the FigLang dataset, but its F1 score was very close to the top baseline. Based on a comprehensive analysis of all evaluation metrics, TED-SCL shows the most competitive and comprehensive performance on different language datasets.

4.5. Representation Visualization

To evaluate the effectiveness of the proposed method in enhancing feature representation, we visualized the hidden layer parameters of each model using t-SNE (t-distributed Stochastic Neighbor Embedding), as shown in Figure 9. The analysis revealed two key findings after incorporating the contrastive learning module and the dual-channel knowledge augmentation module: 1) The feature distributions of toxic euphemism comments and non-toxic comments became

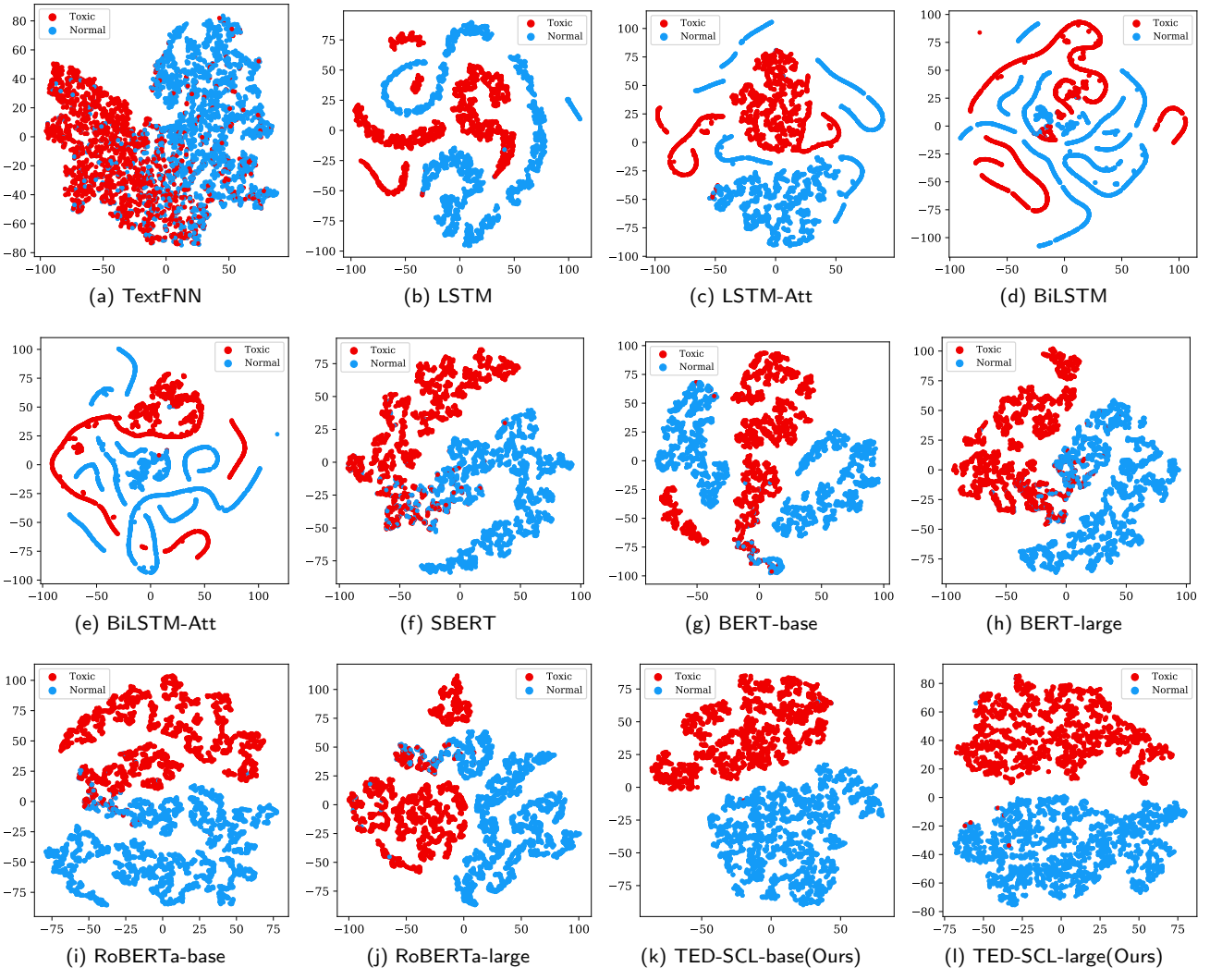


Figure 9: The results of the experiment on representation learning. The representation vectors of toxic and non-toxic samples are plotted on a two-dimensional plane after dimensionality reduction using t-SNE. Red and blue represent the distributions of toxic and non-toxic samples, respectively. The experiment demonstrates that TED-SCL can significantly optimize the distribution of toxic euphemisms and improve detection performance.

more distinct, improving the model's ability to separate these classes. 2) Features related to each target category formed tighter clusters in the feature space, indicating improved learning and alignment with their respective classes.

Baselines perform worse than TED-SCL in extracting toxic euphemism embeddings. They struggle to capture the implicit toxic semantics in euphemistic contexts, resulting in lower classifier accuracy. In contrast, the dual-channel knowledge enhancement module effectively integrates background knowledge of toxic euphemisms, improving the model's performance in understanding and extracting PTETs.

4.6. Ablation Study

Our proposed TED-SCL has two main modules: contrastive learning and dual channel augmentation network. The dual channel knowledge augmentation module includes two channels and a shared attention layer to fuse background knowledge. In the sentiment analysis of the TE-Dataset, it shows that the sentiment feature is useful for classifiers. To evaluate the effectiveness of each component and feature, we conducted a structure ablation experiment and feature ablation experiment.

Table 11
The results of experiment on the structure ablation of TED-SCL. "CL" represents enhanced contrastive learning, "DC" represents dual channel network, and "SA" represents shared attention network.

Methods	ACC.(%)	Tox.			N-Tox.		
		Pre.	Rec.	F1.	N-Pre.	N-Rec.	N-F1.
w/o CL	90.80	88.21	91.07	89.47	92.76	90.72	91.64
w/o DC	92.67	91.32	92.50	91.91	93.80	92.81	93.30
w/o SA	93.09	91.50	**93.32**	92.40	**94.45**	92.91	93.67
TED-SCL	**93.46**	**92.64**	92.85	**92.74**	94.14	**93.96**	**94.05**

Table 12
The results of experiment on the feature ablation of TED-SCL. Tex. represents semantic embeddings of comments, Sen. Represents sentiment feature of comments, Eup. Represents annotation about toxic euphemisms of comments.

Methods	Features			ACC.(%)	Tox.			N-Tox.		
	Tex.	Sen.	Eup.		Pre.	Rec.	F1.	N-Pre.	N-Rec.	N-F1.
TED-SCL-base	✓		✓	93.52	92.50	93.15	92.82	94.36	93.82	94.09
	✓	✓	✓	93.46	92.64	92.85	92.74	94.14	93.96	94.05
TED-SCL-large	✓		✓	93.65	93.02	93.23	93.12	94.44	94.25	94.34
	✓	✓	✓	**93.94**	**93.09**	**93.36**	**93.23**	**94.59**	**94.36**	**94.48**

4.6.1. Structure Ablation

To evaluate the performance of each module components, in this experiment, we removed each component from the whole framework separately, and then repeated 10 times training and testing. The results of the experiment are shown in Table 11:

Firstly, replacing the enhanced encoder with a base RoBERTa model, without CL (contrastive learning), resulted in a noticeable decline in overall performance, demonstrating that CL enhances the model's ability to represent toxic euphemism vectors effectively. Secondly, the removal of the DC (Dual Channel) and SA (Shared Attention) layers from the dual channel knowledge augmentation module led to a decline in various metrics, underscoring the effectiveness of the DC and SA layers in aggregating and utilizing deep background knowledge and sentiment signals.

When the contrastive learning (CL), dual-channel (DC), or shared attention (SA) modules were removed, both precision and recall showed significant decreases. These results demonstrate that the combination of these three structures enhances the model's ability to leverage correlations among different features, effectively distinguishing between toxic euphemistic comments and normal ones, thereby improving detection performance. Overall, each component of the proposed detection method contributes substantially to the detection performance.

4.6.2. Features Ablation

Since the dual channel knowledge augmentation module includes euphemistic meaning annotation feature, the feature effectiveness analysis experiment only focused on sentiment features. The experimental results are summarized in Table 12. When the model used the sentiment feature, it showed better accuracy and F1 score. This suggests that adding more feature types can improve the detection of toxic euphemisms.

4.7. Performance and Parameters Analysis

4.7.1. Real-time Performance

Real-time content moderation is a critical application of toxic content detection systems. To evaluate our proposed framework's potential for real-time content moderation, we compared TED-SCL to other advanced baselines in scalability, real-time performance, and computational requirements. MIT (Maximum Input Tokens) represents the longest text the model can process in a single input, determined by the model architecture and hardware limitations. MOT (Maximum Output Tokens) refers to the number of tokens the model can generate in a single output. AL (Average Latency) is the average time required to process 1,000 tokens, calculated based on the total processing time and the number of tokens handled. RPS (Requests Per Second) is the maximum number of requests the model can process

Table 13
The analysis of the model performance on baselines for real-time content moderation. "MIT" represents maximum input tokens. "MOT" represents the maximum output tokens. "AL" represents average latency of processing, and its unit is ms/1,000 tokens. "RPS" represents the number of maximum requests per second, and its unit is requests/s. "MU" represents memory usage of the running model. "MS" represents the model size of parameters. "LT" represents the time required to load and initialize the model. The unit of "Cost" is $\$/token$. The '*' represents open source and free for usage.

Methods	Scalability		Real Time		Computational Requirements			
	MIT	MOT	AL	RPS	MU	MS	LT	Cost
BERT-base (Devlin et al., 2019)	512	512	10^{-1}	-	2-3 GB	0.11 B	1.78 s	*
BERT-large (Devlin et al., 2019)	512	512	10^{1}	-	8-10 GB	0.34 B	5.35 s	*
RoBERTa-base (Cui et al., 2020)	512	512	10^{-1}	-	2-3 GB	0.12 B	3.58 s	*
RoBERTa-large (Cui et al., 2020)	512	512	10^{1}	-	8-10 GB	0.35 B	6.68 s	*
GPT-4o (OpenAI, 2023)	128,000	4,096	10^{2}	100-120	-	-	-	$5 - 20 * 10^{-6}$
GPT-4-Turbo (OpenAI, 2024)	128,000	4,096	10^{2}	10-20	-	-	-	$10 - 30 * 10^{-6}$
GPT-3.5-Turbo	16,385	4,096	10^{3}	60-80	-	-	-	$1.5 - 3.0 * 10^{-6}$
QWen2-7B (Yang et al., 2024)	128,000	6,144	10^{3}	200-300	-	7 B	-	$0.1 - 0.3 * 10^{-6}$
QWen2-72B (Yang et al., 2024)	128,000	6,144	10^{3}	200-300	-	72 B	-	$0.5 - 1.5 * 10^{-6}$
Llama3-8B (Touvron et al., 2023)	8,000	8,000	10^{3}	-	20-30 GB	8 B	15-20 s	*
Llama3-70B (Touvron et al., 2023)	8,000	8,000	10^{3}	-	70-80 GB	70 B	20-30 s	*
ChatGLM3-6B (Du et al., 2022)	7,500	7,500	10^{3}	-	15-20 GB	6 B	20-30 s	*
TED-SCL-base (Ours)	512	512	10^{-1}	-	3.5 GB	0.10 B	2.35 s	*
TED-SCL-large (Ours)	512	512	10^{1}	-	8.9 GB	0.23 B	5.18 s	*

per second, reflecting its throughput under real-time conditions. MU (Memory Usage) is the peak memory consumed during model execution, while MS (Model Size) refers to the storage space occupied by the model's parameters, typically reported in megabytes or gigabytes. LT (Loading Time) is the time needed to load and initialize the model before processing, determined by the actual startup duration. Lastly, Cost quantifies the expense of generating each token, calculated as the total operational cost divided by the number of tokens produced, providing insights into the financial efficiency of the model.

In Table 13, our proposed method outperforms nearly all advanced LLMs in real-time performance, while significantly reducing computational resource requirements. Compared to traditional baselines, our model offers similar scalability and real-time performance, with slightly lower resource consumption. In terms of MI, LLMs outperform both traditional baselines and TED-SCL. However, in real-time content moderation, comments longer than 128 tokens are relatively rare. Therefore, the MIT and MOT of TED-SCL are sufficient for real-time content moderation of toxic euphemisms. Although LLMs generally have better generalization than TED-SCL in Table 8 and Table 9, TED-SCL has lower AL. Additionally, TED-SCL performs slightly better than other baselines in terms of MU and LT. As a result, TED-SCL has stronger real-time content moderation performance in experiments.

4.7.2. Hyperparameter

The key hyperparameters for the fine-tuning of contrastive learning are the number of epochs, batch size, and maximum sequence length. To find the best hyperparameters for training with contrastive learning, we used a fine-tuning feature extraction model designed for this purpose. We fine-tuned our enhanced encoder by contrastive learning and test it on the TE-Dataset.

As shown in Figure 10a, the model works best when Fine-tune Epochs is set to 10. If this value is too small, the model doesn't learn enough about the differences between features, making it hard for the classifier to perform well. On the other hand, if the value is too large, the features become too spread out, and the classifier struggles to handle similar comments, reducing accuracy. In Figure 10b, the best performance occurs with a batch size of 96. If the batch size is too small, the model may overfit, meaning it focuses too much on specific details and misses the bigger picture. If the batch size is too large, the training can get stuck, and the model fails to learn good representations, which affects the classifier's performance. Figure 10c shows that a Max Sequence Length of 48 gives the best results. If the sequence length is too short, the input comments don't provide enough useful information, and the model can't represent euphemisms accurately. If the sequence length is too long, irrelevant or noisy details are added, which confuses the

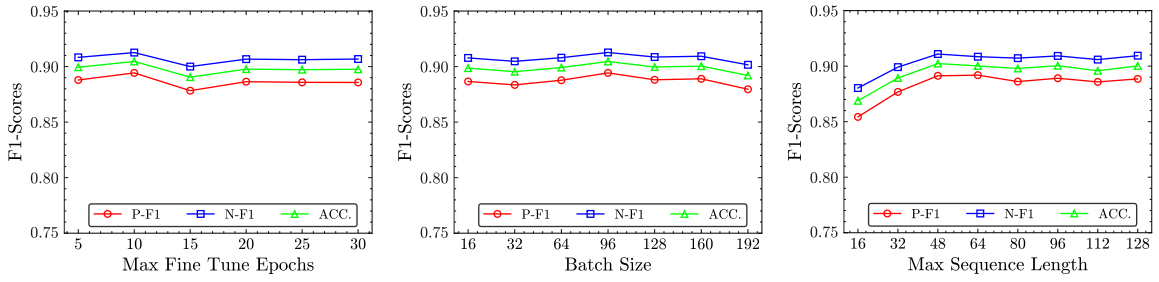


Figure 10: The results of experiment on hyperparameter. Figure 10a illustrates the optimal value for tuning the max fine-tune epochs hyperparameter. Figure 10b shows the best value for adjusting the batch size hyperparameter during model training. Figure 10c presents the ideal value for determining the max sequence length hyperparameter for model training.

Table 14
Toxic euphemism comments (According to the TE-Dataset) classified as non-toxic comments by TED-SCL.

Euphemisms (connotation)	comments
animals (black people) 动物（黑人）	Case 1. I can only say, Humans and animals cannot mix together. 案例1.我只能说，人跟动物就不能混在一起
Han Nan (Korean males) HanNan (韩男)	Case 2. How do you still care about Han Nan, but eyes look really weird. 案例2.怎么还管到HanNan了，不过眼睛看起来确实怪怪的。
Stick (Koreans&Korea) 棒子&棒（韩国人&韩国）	Case 3. Although extremely, particularly, damn, despise stick[nauseated emoji ×3], if China, Japan, and Korea can continue to cooperate friendly and multilaterally, perhaps East Asia could become a miracle. 案例3.虽然非常特别无比巨他妈的讨厌棒子[恶心表情 ×3]，但中日棒能一直友好多边合作的话，说不定东亚能成为奇迹。

model and reduces performance. So, it's important to choose the right sequence length to make the model perform well.

4.8. Case Study

During the detection process, we identified several false positive samples and false negative samples. Manual analysis of these cases revealed certain limitations of our method.

As shown in Table 14, the TED-SCL misclassified toxic euphemisms including "black people," "bad people," and "animals" as harmless euphemisms. This shows that our method relies on contextual and background explanations of euphemisms, and it poses problems in identifying toxic euphemism comments without sufficient context. Given that toxic euphemisms often involve semantic ambiguity, when it comes to extremely ambiguous comments in specific contexts, it is important for classifiers to recognize various types of information. For example, in Case 3, TED-SCL misclassified toxic euphemisms with images or emojis as non-toxic. This issue shows that our method only relies on comments, without enough consideration of non-textual information, such as images, emojis, and videos.

As shown in Table 15, TED-SCL incorrectly classified non-toxic comments that express disdain for toy dolls as toxic. In fact, this is because that the non-toxic comments contain numerous negative words, misleading the model to classify them as offensive speech towards ordinary people. In Case 5, TED-SCL also incorrectly classified self-deprecating text as toxic, due to the confusing potential toxic euphemisms, like "very short" and "low education level". These two cases show that our proposed method can not understand the underlying relationship between toxic euphemisms and sarcasm or mockery. In addition, in Case 6, TED-SCL incorrectly classified non-toxic comments describing the appearance of a Korean person as toxic euphemisms. Although potential toxic euphemism terms may be present, comments without clear intent in their context are generally harmless, except in specific situations.

Table 15
Non-toxic comments (According to the TE-Dataset) classified as the toxic euphemism comments by TED-SCL.

Euphemisms (connotation)	Comments
none	Case 4. I've never had a good feeling towards dolls, especially puppets, toy figures, and the ones with button eyes. Anyone else share this feeling? [crying emoji x2]
无	案例4.从来对娃娃没有好感，特别是木偶啊，玩具小人什么的，最害怕那种眼睛是纽扣做的布娃娃，有没有同感的？[哭泣表情x2]
China aluminum (Chinese females)	Case 5. Sorry, but we, as short-statured, low-educated, and born under unfavorable circumstances China aluminum, are not up to your standards.
国铝（中国女性）	案例5.不好意思，我们身高矮，学历不高，出生也不好的国铝配不上您了。
black devils (black people)	Case 6. Bullet comments should not say that all black devils are bad people, because there are good and bad people everywhere.
黑佬（黑人）	案例6.弹幕不应该说黑佬都是坏人，因为都有好人和坏人呀。

5. Implications

Our purpose of proposing TED-SCL is to detect the toxic euphemism comments on social networks, which attack others by distorting the healthy ordinary words through techniques like variant words, metaphor, or sarcasm (Lu et al., 2023; Sap, Swayamdipta and et al., 2022; Pavlopoulos, Laugier and Xenos, 2022). There are key points that need to be considered: (1) Toxic euphemism comment detection is challenging due to the lack of high-standard labeled datasets and annotations for potential toxic euphemisms. (2) The majority of the words in the toxic euphemism comment are healthy and friendly Superficially (Naim, Hossain and et al., 2022; Elbasani and Kim, 2022; Jia et al., 2023; Gupta et al., 2023; de Paula et al., 2023), which makes it very hard for classifiers to extract and capture the underlying toxic meaning of the toxic euphemism comments. In fact, majority of potential toxic euphemism comments have specific background knowledge or meme. (3) The vectors of toxic euphemism samples are very similar to healthy samples (Yuan et al., 2018; Lao et al., 2021; Ke et al., 2022; Wang et al., 2022; Hu et al., 2024b), which makes the encoder harder to extract real and accurate embeddings from toxic euphemism comments and non-toxic comments due to the vectors space collapse.

Thus, a toxic euphemism dataset and a novel method are necessary for detecting toxic euphemisms. Our work constructed the TE-Dataset , which includes potential toxic euphemisms and corresponding annotations, and we proposed a toxic euphemism detection framework (TED-SCL) based on a contrastive learning module and a dual-channel knowledge augmentation module. Existing research usually ignores the potential euphemistic toxicity and easily fails to detect toxic euphemisms with deep background knowledge through deceptive tricks, such as homophonic words, abbreviations, and metaphors. Due to the lack of background knowledge and comprehension of euphemistic semantics, baselines of toxicity classifiers failed to detect deep toxic euphemisms in experiments. Our research shows that the combination of contrastive learning and dual-channel networks can work together. In Figure 4, contrastive learning helps to reduce the collapse of the vector space of toxic euphemisms, and also increases differentiation between the original sample and replaced sample* in dual-channel networks. On the other hand, dual-channel networks amplify the effectiveness of distancing close embeddings of positive and negative samples in contrastive layers, and also improve the background information comprehension of PTETs. These two modules work together and achieve higher precision, recall , and F-1 scores in comprehensive experiments.

6. Conclusion

We constructed the TE-Dataset and annotated each potential euphemistic toxic term. And we also proposed the TED-SCL framework that combines contrastive learning (self-supervised training) and dual channel augmentation network (supervised training) to improve detecting performance of toxic euphemism comments on social networks. The TED-SCL can classify deceptive toxic euphemism comments accurately and detect deep euphemistic toxicity of comments on social networks.

Annotated TE-Dataset Construction. This paper defines PTET and collects corpus from Bilibili. Relying on existing research (Lu et al., 2023), we manually annotated and constructed a Chinese Toxic Euphemism Dataset (TE-Dataset), including over 400 PTEs and 18,971 comments. The topics of TE-Dataset include six categories: "种

族" (Race), "性别" (Gender), "日常" (General), "色情" (Sexism), "性少数群体" (LGBT), and "地区" (Region). Additionally, we made a dictionary of PTEs and it has hundreds of potential euphemistic terms and corresponding annotation. As specific and valuable dataset, it can help the research of detecting toxic euphemisms and implicit toxic language in the future.

Enhanced Toxic Euphemism Features. This paper proposed the contrastive learning module based on different data augmentation modes and contrastive layers. On the on hand, the contrastive learning module helps to reduce the vector collapse problem of toxic euphemism embedding and improve representing performance of encoder. On the other hand, the module distanced toxic euphemism samples from healthy euphemism samples in vector space and achieved existing SOTA of detecting toxic euphemism comments.

Toxic Euphemism Knowledge Augmentation. This paper proposed the dual channel knowledge augmentation module based on dual channel network and shared attention. The results of comprehensive experiments illustrated the generalization and robustness of this module, which outperforms other baselines. This module fused PTETs knowledge by replacing operation and enhanced the comprehension capacity of the model to improve zero-shot detecting performance. Due to the balanced parameters size and detecting efficiency, the dual channel Long short-term Memory network structure can extract the toxic euphemistic features more easily, by the shallow semantic channel and the deep semantic channel.

Limitations. There are still some limitations of our work: (1) TED-SCL framework designed for detecting toxic euphemisms by us need additional manual work to construct the PTETD. Results of experiments have demonstrated the preliminary cross-domain (topics and datasets) applicability of TED-SCL, however we think it can enhance adaptations to new toxic euphemisms to improve generality; (2) Existing research only considered textual data, but there are different types of toxic and euphemistic content on social networks, such as images, videos, or audio. We will research how to integrate multi-modal information with PTETs to identify toxic euphemisms more effectively.

7. Acknowledgments

This work was supported by Key Research and Development Program of Science and Technology Department of Sichuan Province under Grant No. 2023YFG0145 and the Fundamental Research Funds for the Central Universities (Grant No. SCU2024D012). Additionally, this work is partly supported by the Science and Engineering Connotation Development Project of Sichuan University (Grant No. 2020SCUNG129). Moreover, this work was also supported by the National Natural Science Foundation of China (Grant No. 62202320), the Fundamental Research Funds for the Central Universities (Grant No. 2023SCU12126).

References

Aljawazeri, J., Jasim, M.N., 2024. Addressing challenges in hate speech detection using bert-based models: A review. Iraqi Journal For Computer Science and Mathematics 5, 1–20.

Ameya, V., Feng, M., Yue, N., 2020. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection, in: Proceedings of the International AAAI Conference on Web and Social Media, Virtual Event, June 8–11, pp. 683–693.

Anjalie, F., Yulia, T., 2020. Unsupervised discovery of implicit gender bias, in: Proceedings of the 17th Conference on Empirical Methods in Natural Language Processing, Virtual Event, November 16-20, p. 596–608.

Bahgat, M., Wilson, S., Magdy, W., 2022. Liwc-ud: Classifying online slang terms into liwc categories, in: Proceedings of the 14th ACM Web Science Conference, Barcelona, Spain, June 22-26, pp. 422–432.

Chaudhary, A., Kolhe, S., Kamal, R., 2016. An improved random forest classifier for multi-class classification. Information Processing in Agriculture 3, 215–222.

Chaves, A.P., Gerosa, M.A., 2021. How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. International Journal of Human–Computer Interaction 37, 729–758.

Chen, J., Shen, D., Chen, W., Yang, D., 2021. Hiddencut: Simple data augmentation for natural language understanding with better generalizability, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, August 1-6, pp. 4380–4390.

Choi, M., Kim, H., Han, B., Xu, N., Lee, K.M., 2020. Channel attention is all you need for video frame interpolation, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, February 7-12, pp. 10663–10671.

Clarke, C., Hall, M., Mittal, G., et al., 2023. Rule by example: Harnessing logical rules for explainable hate speech detection, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Toronto, Canada, July 9-14,, pp. 364–376.

Cui, Y., Che, W., Liu, T., Qin, B., Wang, S., Hu, G., 2020. Revisiting pre-trained models for chinese natural language processing. arXiv preprint arXiv:2004.13922 .

Dai, W., Tao, J., Yan, X., Feng, Z., Chen, J., 2023. Addressing unintended bias in toxicity detection: An lstm and attention-based approach, in: Proceedings of the 5th International Conference on Artificial Intelligence and Computer Applications, Dalian, China, November 28-30, pp. 375–379.

Deng, J., Zhou, J., et al., H.S., 2022. COLD: A benchmark for chinese offensive language detection, in: Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, December 7-11, pp. 11580–11599.

Dessì, D., Recupero, D.R., Sack, H., 2021. An assessment of deep learning models and word embeddings for toxicity detection within online textual comments. Electronics 10, 779–787.

Devlin, J., Kenton, Toutanova, L.K., 2019. Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 18th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Albuquerque, New Mexico, USA, July 17-19, Minneapolis, Minnesota. pp. 2–21.

Du, Z., Qian, Y., Liu, X., 2022. GLM: general language model pretraining with autoregressive blank infilling, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 22-27, pp. 320–335.

Dubey, K., Nair, R., Khan, M.U.e.a., 2020. Toxic comment detection using lstm, in: Proceedings of the 3rd International Conference on Advances in Electronics, Computers and Communications, Bengaluru, India, December 11-12, pp. 1–8.

Elbasani, E., Kim, J.D., 2022. Amr-cnn: Abstract meaning representation with convolution neural network for toxic content detection. Journal of Web Engineering 1, 677–692.

Eronen, J., Ptaszynski, M., Masui, F., Arata, M., Leliwa, G., Wroczynski, M., 2022. Transfer language selection for zero-shot cross-lingual abusive language detection. Information Processing and Management 59, 102981–102992.

Felt, C., Riloff, E., 2020. Recognizing euphemisms and dysphemisms using sentiment analysis, in: Proceedings of the 2th Workshop on Figurative Language Processing, Online, July 9-13, pp. 136–145.

Fernandez, E., Winata, M.G., Fasya, F.H.e.a., 2022. Improving indobert for sentiment analysis on indonesian stock trader slang language, in: Proceedings of the 5th IEEE International Conference on Internet of Things and Intelligence Systems, Bali, Indonesia, November 28-30, pp. 240–244.

Fortuna, P., Soler-Company, J., Wanner, L., 2021. How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets? Information Processing and Management 58, 102524–102541.

Frenda, S., Cignarella, A.T., Basile, V.e.a., 2022. The unbearable hurtfulness of sarcasm. Expert Systems with Applications 193, 116398–116416.

Gavidia, M., Lee, P., et al., A.F., 2022. Cats are fuzzy pets: A corpus and analysis of potentially euphemistic terms, in: Proceedings of the 13th Language Resources and Evaluation Conference, Marseille, France, June 20-25, pp. 2658–2671.

Gaydhani, A., Doma, V., Kendre, S., et al., 2018. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. arXiv preprint arXiv:1809.08651 .

Gu, Y., Luo, X., Yang, M., 2024. Incomplete observations bias suppression for abductive natural language inference, in: Proceedings of the 49th IEEE International Conference on Acoustics, Speech and Signal Processing, Seoul, Korea, April 14-19, pp. 10046–10050.

Gupta, S., Lee, S., De-Arteaga, M.e.a., 2023. Same same, but different: Conditional multi-task learning for demographic-specific toxicity detection, in: Proceedings of the 15th ACM Web Conference, Austin, Texas, USA, April 30 - May 4, pp. 3689–3700.

Hada, T., Sei, Y., et al., Y.T., 2023. Detection of compound-type dark jargons using similar words, in: Proceedings of the 15th International Conference on Agents and Artificial Intelligence, Lisbon, Portugal, February 22-24, pp. 427–437.

He, W., Vieira, T.K., Garcia, M.e.a., 2024. Investigating idiomaticity in word representations. Computational Linguistics , 1–48.

Hou, Y., Wang, H., Wang, H., 2022. Identification of chinese dark jargons in telegram underground markets using context-oriented and linguistic features. Information Processing and Management. 59, 103033–103053.

Hu, Y., Li, J., Wang, T.e.a., 2024a. A unified generative framework for bilingual euphemism detection and identification, in: Findings of the 62nd Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, August 11-16, pp. 6753–6766.

Hu, Y., Li, J., Wu, M.e.a., 2024b. Uncovering and mitigating the hidden chasm: A study on the text-text domain gap in euphemism identification, in: Proceedings of the 38the AAAI Conference on Artificial Intelligence, Vancouver, Canada, February 20–27, pp. 18270–18278.

Jessica, L., 2022. Leveraging world knowledge in implicit hate speech detection, in: Proceedings of the 2nd Workshop on NLP for Positive Impact, Abu Dhabi, United Arab Emirates, December 7-8, p. 31–39.

Jia, Y., Wu, W., et al., F.C., 2023. In-game toxic language detection: Shared task and attention residuals, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence, Washington, DC, USA, February 7-14, pp. 16238–16239.

Jiang, A., Yang, X., Liu, Y., Zubiaga, A., 2022. SWSR: A chinese dataset and lexicon for online sexism detection. Online Social Networks Media 27, 100182–100226.

Jiawen, D., Zhuang, C., Hao, S., 2023. Enhancing offensive language detection with data augmentation and knowledge distillation. Research 6, 1–12.

Kapron-King, A., Xu, Y., 2021. A diachronic evaluation of gender asymmetry in euphemism. arXiv preprint arXiv:2106.02083 .

Ke, L., Chen, X., Wang, H., 2022. An unsupervised detection framework for chinese jargons in the darknet, in: Candan, K.S., Liu, H., Akoglu, L., Dong, X.L., Tang, J. (Eds.), Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Online, AZ, USA, February 21-25, pp. 458–466.

Keh, S.S., Bharadwaj, R.K., Liu, E.e.a., 2022. Eureka: Euphemism recognition enhanced through knn-based methods and augmentation. arXiv preprint arXiv:2210.12846 .

Kesen, I., Erdem, A., et al., E.E., 2022. Detecting euphemisms with literal descriptions and visual imagery. CoRR abs/2211.04576.

Kravchenko, H., et al., 2023. IT slang analysis system, in: Proceedings of the 11th Modern Machine Learning Technologies and Data Science Workshop, Lviv, Ukraine, June 3-7, pp. 561–571.

Lao, Y., Zhang, C., et al., Y.W., 2021. Detecting and finding the true meaning of jargons, in: Proceedings of the 7th International Conference on Frontiers of Educational Technologies, Bangkok, Thailand, June 4-7, pp. 45–50.

Le, K.M., Pham, T., et al., T.Q., 2024. LAMPAT: low-rank adaption for multilingual paraphrasing using adversarial training, in: Proceedings of the 38th Conference on Artificial Intelligence, 36th Conference on Innovative Applications of Artificial Intelligence, 40th Symposium on Educational Advances in Artificial Intelligence, Vancouver, Canada, February 20-27, pp. 18435–18443.

Lee, H., Hudson, D.A., Lee, K., Manning, C.D., 2020. Slm: Learning a discourse language representation with sentence unshuffling, in: Proceedings of the 17th Conference on Empirical Methods in Natural Language Processing, Online, November 16-20, pp. 1551–1562.

Lee, P., Trujillo, A.C., Plancarte, D.C.e.a., 2024. Meds for pets: Multilingual euphemism disambiguation for potentially euphemistic terms. arXiv preprint arXiv:2401.14526 .

Li, J., Du, T., Ji, S.e.a., 2020a. {TextShield}: Robust text classification based on multimodal embedding and neural machine translation, in: Proceedings of the 29th USENIX Security Symposium, Seattle, Washington, USA, August 13–15, pp. 1381–1398.

Li, J., Sun, A., Han, J.e.a., 2020b. A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering 34, 50–70.

Liu, D., Deng, H., Cheng, X., Ren, Q., Wang, K., Zhang, Q., 2024. Towards the difficulty for a deep neural network to learn concepts of different complexities. Advances in Neural Information Processing Systems 36, 41283–41304.

Lu, J., Xu, B., et al., X.Z., 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, Toronto, Canada, July 9-14, pp. 16235–16250.

Madaan, A., Setlur, A., et al., T.P., 2020. Politeness transfer: A tag and generate approach, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, July 5-10, pp. 1869–1881.

Madhu, H., Satapara, S., Modha, S.e.a., 2023. Detecting offensive speech in conversational code-mixed dialogue on social media: A contextual dataset and benchmark experiments. Expert Systems with Applications 215, 119342–119361.

Mathew, B., et al., 2021. Hatexplain: A benchmark dataset for explainable hate speech detection, in: Proceedings of the 35th AAAI conference on artificial intelligence, Virtual Event, Feburary 2-9, pp. 14867–14875.

Matsumoto, K., Ren, F., Matsuoka, M.e.a., 2019. Slang feature extraction by analysing topic change on social media. CAAI Transactions on Intelligence Technology 4, 64–71.

Naim, J., Hossain, T., et al., F.T., 2022. Leveraging fusion of sequence tagging models for toxic spans detection. Neurocomputing 500, 688–702.

Nelatoori, K.B., Kommanti, H.B., 2023. Multi-task learning for toxic comment classification and rationale extraction. Journal of Intelligent Information Systems 60, 495–519.

Neog, M., Baruah, N., 2024. A deep learning framework for assamese toxic comment detection: Leveraging lstm and bilstm models with attention mechanism. Advances in Data-Driven Computing and Intelligent Systems 2, 485–498.

OpenAI, 2023. GPT-4o technical report. CoRR abs/2303.08774 .

OpenAI, 2024. GPT-4-Turbo introduction. https://openai.com/index/hello-gpt-4o/ .

de Paula, A.F.M., Rosso, P., Spina, D., 2023. Mitigating negative transfer with task awareness for sexism, hate speech, and toxic language detection, in: Proceedings of the 23th International Joint Conference on Neural Networks, Gold Coast, Australia, June 18-23, pp. 1–8.

Pavlopoulos, J., Laugier, L., Xenos, A.e.a., 2022. From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 22-26, pp. 3721–3734.

Pei, Z., Sun, Z., Xu, Y., 2019. Slang detection and identification, in: Proceedings of the 23rd conference on computational natural language learning, Hong Kong, China, November 3-4, pp. 881–889.

Pérez, J.M., Luque, F.M., 2019. Atalaya at semeval 2019 task 5: Robust embeddings for tweet classification, in: Proceedings of the 13th international workshop on semantic evaluation, Minneapolis, Minnesota, USA, June 17-19, pp. 64–69.

Portnoff, R.S., Afroz, S., et al., G.D., 2017. Tools for automated analysis of cybercriminal markets, in: Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, April 3-7, pp. 657–666.

Rababah, H.A., 2014. The translatability and use of x-phemism expressions (x-phemization): Euphemisms, dysphemisms and orthophemisms in the medical discourse. Studies in Literature and Language 9, 229–240.

Sap, M., Swayamdipta, S., et al., L.V., 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, United States, July 10-15, 2022, pp. 5884–5906.

Song, H., Hong, J., Jung, C.e.a., 2024. Detecting offensive language in an open chatbot platform, in: Proceedings of the 17th Joint International Conference on Computational Linguistics, Language Resources and Evaluation, Torino, Italia, May 20-25, pp. 4760–4771.

Sreyan, G., Manan, S., Purva, C., 2023. Cosyn: Detecting implicit hate speech in online conversations using a context synergized hyperbolic network, in: Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing, Singapore, December 7-11, p. 6159–6173.

Sumanth, P., Samiuddin, S., Jamal, K.e.a., 2022. Toxic speech classification using machine learning algorithms, in: Proceedings of the 1st International Conference on Electronic Systems and Intelligent Computing, Chennai, India, April 22-23, pp. 257–263.

Sun, Z., Zemel, R., Xu, Y., 2021. A computational framework for slang generation. Transactions of the Association for Computational Linguistics 9, 462–478.

Touvron, H., Lavril, T., et al., G.I., 2023. Llama: Open and efficient foundation language models. CoRR abs/2302.13971.

Wan, S., Pan, S., Yang, J.e.a., 2021. Contrastive and generative graph convolutional networks for graph-based semi-supervised learning, in: Proceedings of the 35th AAAI conference on artificial intelligence, Online, February 2-9, pp. 10049–10057.

Wang, H., Hou, Y., Wang, H., 2021. A novel framework of identifying chinese jargons for telegram underground markets, in: Proceedings of the 30th International Conference on Computer Communications and Networks, Athens, Greece, July 19-22, pp. 1–9.

Wang, W., Huang, J., et al., C.C., 2023. Validating multimedia content moderation software via semantic fusion, in: Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis, Seattle, WA, USA, July 17-21, pp. 576–588.

Wang, Y., Su, H., et al., Y.W., 2022. SICM: A supervised-based identification and classification model for chinese jargons using feature adapter enhanced BERT, in: Proceedings of the 19th Pacific Rim International Conference on Artificial Intelligence, Shanghai, China, November 10-13, pp. 297–308.

Wiriyathammabhum, P., 2022. TEDB system description to a shared task on euphemism detection 2022. CoRR abs/2301.06602.

Wyner, A.J., Olson, M., Bleich, J., Mease, D., 2017. Explaining the success of adaboost and random forests as interpolating classifiers. Journal of Machine Learning Research 18, 1–33.

Xiaochuang, H., Yulia, T., 2020. Fortifying toxic speech detectors against veiled toxicity, in: Proceedings of the 17th Conference on Empirical Methods in Natural Language Processing, Virtual Event, November 16-20, p. 7732–7739.

Yang, A., Yang, B., Hui, B.e.a., 2024. Qwen2 technical report. arXiv preprint arXiv:2407.10671 .

Yang, H., Ma, X., et al., K.D., 2017. How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy, in: Proceedings of the 38th IEEE Symposium on Security and Privacy, San Jose, CA, USA, May 22-26, pp. 751–769.

Youngwook, K., Shinwoo, P., Yo-Sub, H., 2022. Generalizable implicit hate speech detection using contrastive learning, in: Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, October 12-17, p. 6667–6679.

Yuan, K., Lu, H., et al., X.L., 2018. Reading thieves' cant: Automatically identifying and understanding dark jargons from cybercrime marketplaces, in: Proceedings of the 27th USENIX Security Symposium, Baltimore, MD, USA, August 15-17, pp. 1027–1041.

Zampieri, M., et al., 2019. Predicting the type and target of offensive posts in social media. arXiv preprint arXiv:1902.09666 .

Zeng, H., Cui, X., 2022. Simclrt: A simple framework for contrastive learning of rumor tracking. Engineering Applications of Artificial Intelligence 110, 104757–104769.

Zhang, C., Benz, P., Imtiaz, T., Kweon, I.S., 2020. Understanding adversarial examples from the mutual influence of images and perturbations, in: Proceedings of the 38th IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA, June 13-19, pp. 14521–14530.

Zhang, J., Wu, Q., Xu, Y., Cao, C., Du, Z., Psounis, K., 2024. Efficient toxic content detection by bootstrapping and distilling large language models, in: Proceedings of the 38th AAAI Conference on Artificial Intelligence, Vancouver, Canada, February 22-25, pp. 21779–21787.

Zhang, Z., Xu, Z.Q.J., 2024. Implicit regularization of dropout. IEEE Transactions on Pattern Analysis and Machine Intelligence 46, 4206–4217.

Zhao, K., Zhang, Y., et al., C.X., 2016. Chinese underground market jargon analysis based on unsupervised learning, in: Proceedings of the 13th IEEE Conference on Intelligence and Security Informatics, Tucson, AZ, USA, September 28-30, pp. 97–102.

Zhou, J., Deng, J., et al., F.M., 2022. Towards identifying social bias in dialog systems: Framework, dataset, and benchmark, in: Proceedings of the Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, Decemember 7-11, pp. 3576–3591.

Zhu, W., Gong, H., Bansal, R.e.a., 2021. Self-supervised euphemism detection and identification for content moderation, in: Proceedings of the 42th IEEE Symposium on Security and Privacy, Online, MAY 24-27, pp. 229–246.