System Report for CCL25-Eval Task 10: Prompt-Driven Large Language Model Merge for Fine-Grained Chinese Hate Speech Detection

Binglin Wu[†], **Jiaxiu Zou**[†], **Xianneng Li**[*]
School of Economics and Management, Dalian University of Technology
xianneng@dlut.edu.cn

Abstract

The proliferation of hate speech on Chinese social media poses urgent societal risks, yet traditional systems struggle to decode context-dependent rhetorical strategies and evolving slang. To bridge this gap, we propose a novel three-stage LLM-based framework: **Prompt Engineering**, **Supervised Fine-tuning**, and **LLM Merging**. First, context-aware prompts are designed to guide LLMs in extracting implicit hate patterns. Next, task-specific features are integrated during supervised fine-tuning to enhance domain adaptation. Finally, merging fine-tuned LLMs improves robustness against out-of-distribution cases. Evaluations on the STATE-ToxiCN benchmark validate the framework's effectiveness, demonstrating superior performance over baseline methods in detecting fine-grained hate speech.

1 Introduction

The rapid growth of social media platforms has led to a global surge in online hate speech, which not only inflicts psychological harm on targeted individuals or groups but also exacerbates social tensions and fuels collective antagonism (Arora et al., 2023). While existing technologies can preliminarily detect explicit hate content (Schmidt and Wiegand, 2017), Chinese hate expressions are often characterized by implicitness, diversity, and context-dependency (Qian et al., 2018). Offensive content may be embedded through metaphors, sarcasm, or indirect references (Fortuna and Nunes, 2018), frequently targeting specific group attributes such as geography, gender, or ethnicity (Mathew et al., 2021). Against this backdrop, **fine-grained hate speech detection** has emerged as a critical research direction to address this issue. It aims to precisely dissect hate elements—such as target entities, arguments, victimized groups, and hate attributes (Vidgen et al., 2021)—from textual data, enabling more accurate identification and regulation of online hate speech.

The core requirement of fine-grained hate speech detection lies in models that can not only recognize explicit offensive lexicons but also infer discriminatory intent from contextual semantics (ElSherief et al., 2021), while strictly adhering to structured output specifications (Pavlopoulos et al., 2020). However, current mainstream models face three critical bottlenecks:(1)**Semantic Complexity**: Traditional rule-based or shallow machine learning methods, as well as directly applied large language models, struggle to accurately capture the implicit and diverse fine-grained semantic features inherent in Chinese hate speech (Talat and Hovy, 2016).(2)**Incomplete Information Extraction**: General-purpose pre-trained models lack targeted attention to hate speech components, resulting in incomplete extraction of critical information.(3)**Generalization Limitations**: Single training paradigms are susceptible to data distribution biases, limiting model generalization in complex scenarios and hindering adaptability to dynamic online environments (Gururangan et al., 2020).

†Equal Contribution
*Corresponding Author

Proceedings of the 24th China National Conference on Computational Linguistics, pages 403-410, Jinan, China, August 11-14, 2025.

To address these challenges, this study proposes a hybrid training framework based on the Qwen2.5-7B-Instruct LLM (Qwen et al., 2025), employing a three-stage optimization strategy. First, Prompt Engineering guides the model to focus on hate speech elements (e.g., victimized group classification and metaphor identification rules) while enforcing structured output through task-oriented templates. Second, Supervised Fine-Tuning (SFT) (Ouyang et al., 2022) enhances the model's ability to parse fine-grained semantics using high-quality annotated data, particularly improving discrimination accuracy for implicit hate expressions. Finally, Model Merging (Matena and Raffel, 2022) innovatively integrates multi-stage models via the LLM Merging method, which sparsifies task vectors by pruning extreme parameters, thereby synthesizing complementary features from different training phases to boost robustness. Experimental results demonstrate stable performance scores of 0.3553 and 0.3555 on preliminary and final test sets, respectively, with over 15% accuracy improvement in hate detection compared to baseline models. The fused model also exhibits exceptional adaptability in complex scenarios such as multi-group attacks and cross-context generalization. This work provides a theoretically innovative and practically valuable technical pathway for Chinese fine-grained hate speech detection, contributing significantly to fostering safer online discourse environments.

2 Methodology

2.1 Framework Overview

The proposed framework comprises three pivotal components: (1) Domain-specific Prompt Engineering, (2) Task-oriented Supervised Fine-tuning, and (3) Dynamic LLM Merge. As illustrated in the hierarchical architecture of the algorithmic framework figure 1, the system operates through phased optimization: prompt engineering guides the model to concentrate on fine-grained hate elements, the supervised fine-tuning phase enhances the model's discriminative capacity for implicit semantic nuances, and model merge enhances both the recognition accuracy and generalization capabilities of the system.

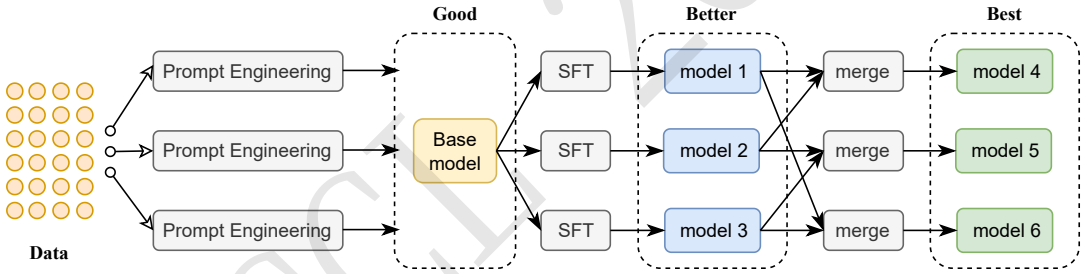

Figure 1: Framework Architecture

2.2 Domain-specific Prompt Strategy

The Prompt Strategy enhances structured output capabilities and fine-grained hate judgment logic through domain-specific prompt template design. Specifically, the prompt template incorporates three core components:

First, it defines clear task objectives by mandating the model to output results following a "four-tuple" structured framework. To reinforce the model's understanding of this format, contextual examples are strategically embedded immediately after defining each field.Second, it embeds explicit definitions of hate speech while establishing contrasting non-targeted content boundaries through dual-directional examples. For instance, the prompt explicitly contrasts hate speech with non-targeted content, clarifying criteria with phrases like "ordinary information without group targeting does not constitute hate speech." This bidirectional guidance reduces false positives by sharpening the model's ability to differentiate subtle boundary cases.Third, it optimizes target group extraction by integrating predefined category explanations that map to common social group attributes. The prompt systematically breaks down each category's defining features, and emphasizes handling overlapping scenarios. This structured approach

ensures the model prioritizes contextually relevant group attributes while mitigating oversights in complex expressions.

Figure 2 is a sample prompt. Through these designs, the prompt shifts the model's focus from generalized semantic analysis to targeted feature extraction governed by structured guidelines, aligning detection logic with fine-grained hate speech components. The domain-specific prompt not only enforces rigorous output formatting but also embeds implicit reasoning pathways for decoding implicit rhetoric.

Chinese:

你是一个专业的中文仇恨言论分析专家，请根据输入的社交媒体文本，精准识别并输出仇恨言论四元组。确保论点与评论对象准确对应，避免简单读取句子中的谓语作为论点。同时，论点必须是一个完整的谓语动作或谓宾结构，不要随意截断。特别注意隐喻和暗示必须结合具体语境进行判断。请严格按照以下规则处理：

#任务说明：1.分析文本中所有可能的评论对象-论点对，每个对生成一个四元组，评论对象是被评价的具体对象(如人、群体、事物),论点 是针对该对象的具体描述片段。2.输出顺序必须为：评论对象|论点|目标群体|是否仇恨3.必须保留原文片段原貌，禁止改写或缩略#字段规范：1.评论对象：被评价的具体对象(无明确对象时填NULL)2.论点：针对该对象的负面评价片段(必须为原文连续片段)3.目标群体分类：-Region:地域歧视(例：攻击河南人/农村人)-Racism:种族/民族歧视(例：攻击黑人/少数民族)-Sexism:性别歧视(例：攻击女性/性别特征)-LGBTQ:性少数群体歧视-others:其他明确群体(例：残疾人/宗教群体)-non-hate:不针对特定群体4.仇恨判定标准：√存在贬低/侮辱特定群体的意图√传播群体负面刻板印象√煽动对特定群体的敌意×普通攻击/无群体指向不算

#输出规范：1.单元组格式：元素间保留空格并用\"|\"连接，结尾加[END]2.多单元组用[SEP]分隔，示例如下：输入：你可真是头蠢驴，这都做不好。输出：你|蠢驴|non-hate|non-hate[END]输入：老黑我是真的讨厌，媚黑的还倒贴。输出：老黑|讨厌|Racism|hate[SEP]媚黑的|倒贴|Racism|hate[END]3.严格保留所有空格和标点符号。

#特殊处理规则：1.隐喻/暗示需结合语境判断(例：\"娘炮\"可能属Sexism)2.多群体攻击需拆分为多个四元组3.非仇恨内容仍需输出四元组(目标群体设为non-hate)4.保持文本片段原始形态，保留表情和标点，不要改写或缩写5.优先提取仇恨内容，非仇恨内容仅在仇恨内容完全提取后才补充。请处理以下输入：

English:

You are a professional Chinese hate speech analysis expert. Please accurately identify and output the hate speech quadruple based on the input social media text. Ensure that the argument corresponds accurately to the comment object, and avoid simply reading the predicate in the sentence as the argument. At the same time, the argument must be a complete predicate action or predicate-object structure, and should not be arbitrarily truncated. Pay special attention that metaphors and implications must be judged in combination with the specific context. Please strictly follow the following rules for processing:

#Task Instructions: 1. Analyze all possible comment object-argument pairs in the text, and generate a quadruple for each pair. The comment object is the specific object being evaluated (such as a person, group, or thing), and the argument is the specific descriptive fragment about that object. 2. The output order must be: Comment Object | Argument | Target Group | Is Hate 3. The original text fragments must be retained in their original form, and no rewriting or abbreviation is allowed. #Field Specifications: 1. Comment Object: The specific object being evaluated (fill in NULL if there is no clear object). 2. Argument: The negative evaluation fragment about the object (must be a continuous fragment from the original text). 3. Target Group Classification: - Region: Regional discrimination (example: attacking Henan people/rural people) - Racism: Racial/ethnic discrimination (example: attacking Black people/ethnic minorities) - Sexism: Gender discrimination (example: attacking women/gender characteristics) - LGBTQ: Discrimination against sexual minority groups - others: Other specific groups (example: people with disabilities/religious groups) - non-hate: Not targeting a specific group. 4. Hate Judgment Criteria: √ There is an intention to belittle/insult a specific group √ Spread negative stereotypes about a group √ Incite hostility towards a specific group × Ordinary attacks/attacks without group orientation are not considered hate.

#Output Specifications: 1. Unit group format: Retain spaces between elements and connect them with '|', and add [END] at the end. 2. Multiple unit groups are separated by [SEP], as shown in the following examples: Input: You are really a stupid donkey, you can't even do this well. Output: You | stupid donkey | non-hate | non-hate [END] Input: I really hate old blacks, and those who fawn over blacks even stoop to flatter. Output: old blacks | hate | Racism | hate [SEP] those who fawn over blacks | stoop to flatter | Racism | hate [END] 3. Strictly retain all spaces and punctuation marks.

#Special Processing Rules: 1. Metaphors/implications need to be judged in combination with the context (example: "sissy" may belong to Sexism). 2. Attacks on multiple groups need to be split into multiple quadruples. 3. Non-hate content still needs to output a quadruple (set the target group as non-hate). 4. Keep the original form of text fragments, retain emoticons and punctuation, and do not rewrite or abbreviate. 5. Prioritize extracting hate content, and non-hate content is only supplemented after all hate content has been completely extracted. Please process the following input:

Figure 2: Sample Prompt

2.3 Task-oriented Supervised Fine-tuning

Given a pre-trained large language model θ_{pre} and a labeled dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ for hate speech detection, full parameter supervised fine-tuning minimizes the loss function:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} [y_i \log p(\hat{y}_i|\mathbf{x}i;\theta) + (1-y_i)\log(1-p(\hat{y}_i|\mathbf{x}i;\theta))] + \lambda|\theta-\theta_{pre}|^2 \quad (1)$$

where θ denotes the complete set of trainable parameters, $\hat{y}_i$ represents the model's predicted probability for the i-th sample, λ serves as the L2 regularization coefficient that governs parameter magnitude constraints to mitigate overfitting. The parameter update rule of the AdamW optimizer is defined as:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{m_t}{\sqrt{v_t}+\epsilon} - \eta\lambda\theta_t \quad (2)$$

where η denotes the learning rate, m_t and v_t represent the exponentially decaying first and second moment estimates of gradients, respectively, and ϵ is a small constant ensuring numerical stability.

2.4 Dynamic Large Language Model Merge

The capabilities learned by LLMs fine-tuned with different prompt strategies exhibit significant variations.Recent studies show merging large language models (LLMs) effectively enhances performance and generalization. For instance, in e-commerce intention recognition, merged models demonstrate stronger robustness when processing noisy multimodal data, significantly improving accuracy in complex scenarios (Li et al., 2025).Building upon the methodologies presented in (Yadav et al., 2023; Davari and

Belilovsky, 2024), we propose a LLM Merging algorithm to integrate these diverse capabilities, with the detailed workflow outlined in Algorithm 1.

Given fine-tuned LLMs $\{\theta_t\}_{t=1}^{n}$ and a base LLM θ_{base}, we first construct corresponding task vectors τ. Based on task vectors $\{\tau_t\}_{t=1}^{n}$, the LLM Merging method proceeds through three sequential steps to achieve parameter merging:

- **Prune:** We partition the model into layers. For each layer, a masking process is implemented to filter out large outliers and minor perturbations, using α and β as thresholds for the right-tail (upper bound) and left-tail (lower bound) distributions, respectively. The resulting layer-specific masks $m_{t,layer}^{\alpha,\beta}$ are aggregated across all layers to generate the final unified mask $m_t^{\alpha,\beta}$. The mask is then applied to the task vector τ_t to derive the refined parameter set $\hat{\tau}_t$, from which we extract the task-specific direction $\hat{\gamma}_t$ and the magnitude of change $\hat{\mu}_t$.

- **Direct:** We construct a directional alignment vector γ_m to resolve sign inconsistencies among corresponding parameters across different models. Specifically, task vectors sharing the same sign direction are aggregated, and the orientation demonstrating the highest cumulative magnitude is selected as the consensus direction.

- **Merge:** For each parameter, we construct the chosen set of task vectors $\mathcal{A}^p$, which only retains the parameter values of the models whose symbolic directions are the same as the consensus direction. Finally, calculate their average values τ_m^p, scale them and then add them to the base parameters to obtain the final merged parameters θ_m.

Algorithm 1 LLM MERGING Procedure.

Input: Fine-tuned LLMs $\{\theta_t\}_{t=1}^{n}$, Initialization θ_{base}, α, β and λ.
Output: Merged LLM θ_m

for all $t \in [1, ..., n]$ **do**
  ▷ Create task vectors.
  $\tau_t = \theta_t - \theta_{\text{base}}$
  ▷ Step 1: Prune redundant vectors.
  for all $layer \in Layers(\theta)$ **do**
    $m_{t,layer}^{\alpha} \leftarrow$ mask_top_k_percent$(k = \alpha)$
    $m_{t,layer}^{\beta} \leftarrow$ mask_bottom_k_percent$(k = \beta)$
    $m_{t,layer}^{\alpha,\beta} \leftarrow$ merge_masks$(m_t^{\alpha}, m_t^{\beta})$
  end for
  $m_t^{\alpha,\beta} \leftarrow$ stack_masks$(\left\{ m_{t,layer}^{\alpha,\beta} \right\}_{layer \in Layers})$
  $\hat{\tau}_t \leftarrow m_t^{\alpha,\beta} \cdot \tau_t$
  $\hat{\gamma}_t \leftarrow \text{sgn}(\hat{\tau}_t)$
  $\hat{\mu}_t \leftarrow |\hat{\tau}_t|$
end for
▷ Step 2: Indicate task directions.
$\gamma_m = \text{sgn}\left(\sum_{t=1}^{n} \hat{\gamma}_t \odot \hat{\mu}_t\right)$
▷ Step 3: Merge chosen task vectors.
for all $p \in [1, ..., d]$ **do**
  $\mathcal{A}^p = \{t \in [n] \mid \hat{\gamma}_t^p = \gamma_m^p\}$
  $\tau_m^p \leftarrow \frac{1}{|\mathcal{A}^p|} \sum_{t \in \mathcal{A}^p} \hat{\tau}_t^p$
end for
Obtain merged checkpoint
$\theta_m \leftarrow \theta_{\text{base}} + \lambda * \tau_m$
return θ_m

3 Experiments and Results

3.1 Dataset

The STATE-ToxiCN dataset (Bai et al., 2025) comprises 8,000 Chinese social media comments (e.g., from Tieba and Zhihu) annotated with fine-grained quadruples (Target — Argument — Targeted Group — Hateful) for hate speech recognition. Each sample captures explicit targets (or NULL), argumentative fragments, affected groups (geographic, race, gender, LGBTQ, other/Non-hate), and binary hate labels, yielding 9,405 quadruples (5,949 hateful, 3,456 non-hate). It supports multi-target annotations via [SEP] separators and enforces full element extraction even for non-hate texts, covering scenarios like racial bias and gender conflicts. Rigorous validation ensures semantic consistency, offering granular supervision for modeling hate speech components beyond sentence-level classification.

3.2 Evaluation Metrics

The evaluation metrics consist of the F1-scores for hard matching and soft matching between the submitted results and the standard answers, as well as the average of these two F1-scores. The calculation method is consistent with the scikit-learn library.

For the hard matching, a predicted four-tuple is considered correctly extracted if and only if each element of the predicted four-tuple is completely identical to the corresponding element in the answer.

For the soft matching, a predicted four-tuple is considered correctly extracted under the following conditions: the ”Targeted Group” and ”Hateful” elements of the predicted four-tuple are completely identical to the corresponding elements in the standard answer, and the string matching degree of the ”Target” and ”Argument” elements between the predicted four-tuple and the standard answer exceeds 50% . The similarity is calculated as:

$$\text{Similarity} = \frac{M \times 2}{\text{len}_{\text{pred}} + \text{len}_{\text{gold}}} \tag{3}$$

where len_{pred} is the length of the predicted four-tuple, len_{gold} is the length of the standard answer, and M is the length of the longest common subsequence between the predicted four-tuple and the standard answer.

The F1-score is calculated as:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

where P is precision and R is recall. These metrics comprehensively evaluate the performance of the model from both strict matching and partial matching perspectives.

3.3 Experiment Setup

The experiments were conducted using 8 NVIDIA RTX 4090 GPUs. During the training phase, we set the global learning rate to 1e-5, with a total batch size of 64, and trained for 8 epochs. To optimize GPU memory utilization, we employed the DeepSpeed Zero-3 Offload strategy to offload partial model parameters to CPU memory and integrated Flash Attention 2.0 to accelerate attention computation.

3.4 Experiment Results

3.4.1 Overall Comparative Analysis

The comprehensive evaluation of various post-training approaches on different Qwen2.5 variants is shown on Table 1. The RFT method underperformed significantly, likely due to inadequate guidance from its rule-based reward mechanism in capturing nuanced hate speech patterns. Notably, the CPT+SFT approach applied to the base model demonstrated competitive performance, outperforming direct SFT on the Instruct variant (0.3379 vs. 0.3436). We hypothesize that extended CPT training with additional domain-specific corpora could further enhance this performance gap.

Our proposed method achieves state-of-the-art results across all three metrics on Test1, with particularly notable improvements in Hard Score, indicating superior detection capability for implicit hate

expressions. Remarkably, without any task-specific adaptation, our framework maintains robust performance on Test2 (Score: 0.3545), demonstrating both methodological effectiveness and generalization capabilities.

Table 1: Overall results on test1

Base Model	Method	Score	Hard Score	Soft Score
Qwen2.5-3B-Instruct	RFT	0.2021	0.1126	0.2915
Qwen2.5-7B-Base	CPT+SFT	0.3379	0.2353	0.4404
Qwen2.5-7B-Instruct	SFT	0.3436	0.2383	0.4489
Qwen2.5-7B-Instruct	Ours	**0.3553**	**0.2504**	**0.4604**

[1] RFT: Reinforcement Fine-tuning based on GRPO Algorithm (Shao et al., 2024)
[2] CPT+SFT: Continue Pre-training using COLD dataset (Deng et al., 2022). The subsequent SFT uses the prompt strategy of ICL+NH+CE.

3.4.2 Effect of Prompt Strategy

The experimental results in Table 2 demonstrate a progressive improvement as prompt strategies are incrementally enhanced. The baseline ICL approach achieved a score of 0.2921, while the most comprehensive strategy combining ICL with Non-Hate examples, Category Explanations, and explicit Judge Criteria attained the highest performance. This 17.6% relative improvement from baseline to the optimal configuration suggests that clarifying detection boundaries through category explanations and judgment criteria significantly enhances model discernment in ambiguous cases. Particularly, the soft score improvement (14.6% increase) indicates enhanced capability to handle nuanced expressions like sarcasm and homophonic substitutions prevalent in Chinese hate speech.

These results emphasize the importance of combining structured detection guidelines with linguistic and cultural awareness in prompt engineering for Chinese hate speech identification.

Table 2: Results of different prompt strategies on test1

Prompt Strategy	Score	Hard Score	Soft Score
ICL	0.2921	0.1926	0.3916
ICL+Non Hate	0.3279	0.2196	0.4362
ICL+NH+Category Explain	0.3340	0.2316	0.4365
ICL+NH+CE+Judge Criteria	**0.3436**	**0.2383**	**0.4489**

[1] ICL: In-context Learning, specify the task requirements and provide examples

3.4.3 Effect of LLM Merge

Results shown in Table 3 reveal significant performance enhancements through LLM Merge. Merging base ICL with its enhanced version (ICL+NH) produced Merge1 (0.3412 score), already surpassing the standalone ICL+NH+CE model (0.3340). Subsequent merging iterations demonstrated compounding benefits, with Merge2 (0.3530) and Merge3 (0.3553) progressively outperforming all individual prompt-engineered models, including the comprehensive ICL+NH+CE+JC configuration (0.3436). This 3.4% improvement from the best single-model to merged models suggests complementary strengths in different detection approaches – where original models might overfit specific patterns, merged versions likely balance categorical understanding from explicit prompts with nuanced judgment capabilities. Notably, the hard score increased 5.1% (0.2383→0.2504) through merging, indicating improved consensus on definitive hate speech cases, while the 2.5% soft score gain (0.4489→0.4602) reflects enhanced handling of ambiguous expressions.

However, diminishing returns between Merge2 (0.3530) and Merge3 (0.3553) suggest a potential limit to current merging strategies' effectiveness, possibly requiring novel fusion techniques for Chinese's context-dependent hate markers like dialectal variations and historical allusion. These results advo-

cate for hybrid approaches combining prompt engineering with model merging to address Chinese hate speech's unique linguistic and cultural complexity.

Table 3: Results of merged models on test1

Model	Score	Hard Score	Soft Score
ICL	0.2921	0.1926	0.3916
ICL+Non Hate	0.3279	0.2196	0.4362
ICL&ICL+NH (Merge1)	**0.3412**	**0.2358**	**0.4467**
ICL+NH+Category Explain	0.3340	0.2316	0.4365
Merge1&ICL+NH+CE (Merge2)	**0.3530**	**0.2497**	**0.4562**
ICL+NH+CE+Judge Criteria	0.3436	0.2383	0.4489
Merge2&ICL+NH+CE+JC (Merge3)	**0.3553**	**0.2504**	**0.4602**

[1] & means a child model merged from parent models

4 Conclusion

This study presents a novel three-stage framework for fine-grained Chinese hate speech detection, integrating prompt engineering, supervised fine-tuning, and LLM merging. Through systematic experimentation on the STATE-ToxiCN benchmark, we demonstrate that our prompt-driven approach significantly enhances LLMs' capability to decode implicit hate patterns through structured semantic decomposition. The LLM merging algorithm effectively synthesizes complementary detection capabilities from different fine-tuned models. The final merged model exhibits robust performance in handling complex scenarios while maintaining generalization capabilities. The results highlight the potential of the merge-based approach in addressing language-specific challenges, contributing to safer and more inclusive online discourse environments.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant 72071029 and 72231010.

References

Arnav Arora, Preslav Nakov, Momchil Hardalov, Sheikh Muhammad Sarwar, Vibha Nayak, Yoan Dinkov, Dimitrina Zlatkova, Kyle Dent, Ameya Bhatawdekar, Guillaume Bouchard, et al. 2023. Detecting harmful content on online platforms: what platforms need vs. where research efforts go. *ACM Computing Surveys*, 56(3):1–17.

Zewen Bai, Yuanyuan Sun, Shengdi Yin, Junyu Lu, Jingjie Zeng, Haohao Zhu, Liang Yang, and Hongfei Lin. 2025. State toxicn: A benchmark for span-level target-aware toxicity extraction in chinese hate speech detection. *arXiv preprint arXiv:2501.15451*.

MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pages 270–287. Springer.

Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. 2022. Cold: A benchmark for chinese offensive language detection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11580–11599.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4):1–30.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July. Association for Computational Linguistics.

Zhipeng Li, Binglin Wu, Yingyi Zhang, Xianneng Li, Kai Li, and Weizhi Chen. 2025. Cusmer: Multimodal intent recognition in customer service via data augment and llm merge. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 3058–3062.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

John Pavlopoulos, Jeffrey Sorensen, Lucas Dixon, Nithum Thain, and Ion Androutsopoulos. 2020. Toxicity detection: Does context really matter? In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4296–4305, Online, July. Association for Computational Linguistics.

Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 118–123, New Orleans, Louisiana, June. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zeerak Talat and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682, Online, August. Association for Computational Linguistics.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115.