404: Civility Not Found? Evaluating the Effectiveness of Small Language Models in Detecting Incivility in GitHub Conversations

Mário Patrício
Federal University of Ceará
Itapajé, Brazil
jose.mariopatricio@alu.ufc.br

Silas Eufrásio
Federal University of Ceará
Itapajé, Brazil
silaseufrasio@alu.ufc.br

Anderson Uchôa
Federal University of Ceará
Itapajé, Brazil
andersonuchoa@ufc.br

Lincoln S. Rocha
Federal University of Ceará
Fortaleza, Brazil
lincoln@dc.ufc.br

Daniel Coutinho
Pontifical Catholic
University of Rio de Janeiro
Rio de Janeiro, Brazil
dcoutinho@inf.puc-rio.br

Juliana Alves Pereira
Pontifical Catholic
University of Rio de Janeiro
Rio de Janeiro, Brazil
juliana@inf.puc-rio.br

Matheus Paixão
State University of Ceará
Fortaleza, Brazil
matheus.paixao@uece.br

Alessandro Garcia
Pontifical Catholic
University of Rio de Janeiro
Rio de Janeiro, Brazil
afgarcia@inf.puc-rio.br

ABSTRACT

Context: Incivility in open-source software (OSS) platforms like GitHub can harm collaboration, discourage contributor participation, and impact code quality. Although current moderation tools based on Machine Learning (ML) and Natural Language Processing (NLP) offer some support, they often struggle to detect nuanced or implicit types of incivility. **Goal:** This study aims to assess the effectiveness of Small Language Models (SLMs) in detecting both coarse-grained (civil vs. uncivil) and fine-grained (specific types) incivility in GitHub conversations (issues and pull requests), and to understand how different prompting strategies influence detection performance. **Method:** We evaluate ten SLMs (3B-14B parameters) across five prompt strategies, on a labeled dataset with more than 6k GitHub conversations. We also compare the best-performing SLMs with five traditional ML models using two text-encoding techniques. **Results:** Our results reveal that SLMs perform well in detecting civil comments, but their effectiveness in detecting uncivil comments depends on model size. Models with 9B+ parameters (e.g., deepseek-r1, gpt-4o-mini) show improved performance on uncivil comments. For the fine-grained granularity, prompting strategy plays a critical role, with *role-based* prompting achieving the best results, particularly for implicit incivility types (e.g., *Irony* and *Mocking*), even when SLMs struggle with these types of incivility. Traditional ML models still perform well in explicit cases like *Threat* and *Insulting*. **Conclusion:** Our findings highlight the effectiveness of SLMs and prompt strategies in enhancing the detection of incivility within collaborative software development settings.

KEYWORDS

small language models, incivility, moderation, GitHub conversations, open-source projects

1 Introduction

In Open-Source Software (OSS) development environments such as GitHub, developers' conversations play a crucial role in driving project progress. These conversations often take place through discussions on pull requests and issues. However, uncivil behaviors can disrupt such communication, including disrespectful or offensive comments [18, 32, 37]. These behaviors may discourage contributors from actively engaging with a project, and may even result in a decline in code quality over time [9, 13, 19]. For instance, a contributor may open a pull request proposing a code refactor, and receive a comment such as *"This is a terrible idea. Clearly, you don't understand how real developers work."*, a response that exemplifies *mocking* incivility. While the technical merits of the proposal could be discussed constructively, the uncivil tone may discourage further contributions, especially from newcomers [31].

Such conversations, even if rare, can have a cascading effect on community engagement and the overall health of any project. In this context, existing moderation tools based on Machine Learning (ML) and Natural Language Processing (NLP) [21, 34] are increasingly being employed to detect signs of incivility in pull requests or issues discussions, but these approaches are often hard to adapt to the nuances of languages [6]. Recent advances in foundation models have opened new possibilities in software engineering [14], yet their use in identifying different types of incivility, especially in GitHub conversations, remains underexplored.

This study addresses this gap by evaluating the effectiveness of ten Small Language Models (SLMs), ranging from 3 to 14b parameters – *DeepSeek-r1:14b*, *DeepSeek-r1:8b*, *Mistral:7b*, *Mistral-nemo:12b*, *Gemma:7b*, *Gemma2:9b*, *Llama3.2:3b*, *Llama3.1:8b*, *Phi4:14b*, and *GPT-4o-mini* - in detecting incivility in GitHub conversations. Specifically, we assess their performance across different levels of incivility – coarse-grained (civil vs. uncivil) and fine-grained, which include nine distinct types, ranging from Impatience to Irony and Mocking – and prompting strategies, including *zero-shot*, *one-shot*, *few-shot*, *auto-Chain-of-Thought (auto-CoT)*, and *role-based* prompting. We also analyze the effect of prompting strategies on SLM performance. Finally, we compare the best-performing SLM configuration against five traditional ML models combined with two NLP text encoding techniques – Bag-of-Words (BoW) and TF-IDF. We summarize our findings as follows:

- SLMs effectively detect civil comments, maintaining high precision, recall, and F1-scores across all SLMs. However, detecting uncivil comments is more challenging and depends on model size. SLMs with 9B+ parameters, like *gemma2:9b*, *mistral-nemo:12b*, and *gpt-4o-mini*, showed more suitable for moderation tasks. Still, SLMs struggle with implicit incivility (e.g., *Irony* and *Mocking*);
- Prompting strategies are reliable for detecting civil comments but vary in effectiveness for uncivil ones. Structured

prompts like *auto-CoT* and especially *role-based* outperform simpler prompts, enhancing detection of fine-grained incivility through guided reasoning and role context across all types of incivility. *Role-based* prompting achieved the best precision (0.05-0.61) and F1-scores (0.06-0.30), while *few-shot* achieved the worst precision and F1 across strategies;

- *GPT-4o-mini*, guided by *role-based* prompting, outperforms traditional ML models across both civil and uncivil comment detection. While *deepseek-r1:14b* generally outperforms ML baselines, models like *Logistic Regression Classifier + BoW* stand out in explicit incivilities such as *Threat*, *Insulting*, and *Entitlement*, suggesting ML models with feature representations can still be highly effective in specific scenarios.

2 Background

This section provides an overview of the theoretical foundation for understanding the key concepts of the study as follows.

SLMs and Prompt Engineering. SLMs are lightweight alternatives to Large Language Models (LLMs), making them particularly attractive for integration into real-time or resource-constrained moderation pipelines, given their faster inference, reduced computational cost, and easier deployment. However, their reduced size also results in a lower capacity for reasoning and knowledge retrieval compared to LLMs [29]. To maximize their effectiveness, prompt engineering plays a crucial role in guiding SLMs to generate high-quality responses despite their constraints. Prompt engineering involves creating task-specific instructions, known as prompts, to improve model performance without modifying its core parameters [36]. Instead of fine-tuning the model, prompts guide pre-trained models to perform downstream tasks by eliciting the desired behavior through carefully crafted instructions [22]. Prompting techniques range from simple to complex, and their effectiveness depends on the task complexity. Sahoo et al. [26] conducted a systematic survey on prompt engineering, classifying various strategies based on their suitability for different domains. Table 1 provides an overview of the common prompting techniques discussed in the literature and employed in our study.

Table 1: Overview of existing prompting techniques

Technique	Description
Zero-shot [22]	This technique eliminates the need for extensive training data, relying instead on carefully crafted prompts that guide the model toward novel tasks based solely on the model's pre-existing knowledge.
One-shot [22]	This technique involves giving a model a single example of a task at the time of use. The model then uses the example to understand and complete a new, similar task, which reduces the need for large amounts of task-specific data.
Few-shot [2]	This technique involves giving a model a few examples of a task at the time of use. The model then uses these examples to understand and complete a new, similar task, which reduces the need for large amounts of task-specific data.
Automatic Chain-of-Thought (auto-CoT) Prompting [35]	Instead of writing manual reasoning demonstrations one by one for each example, auto-CoT leverages a simple prompt like "Let's think step by step" to facilitate step-by-step thinking before answering a question.
Role-based Prompting [17]	This technique involves providing the model with a specific role in the prompt. The assigned role offers context about the LLM's identity and background, enabling it to generate more natural, in-character responses tailored to that role.

Incivility in GitHub Conversations. Open-source software (OSS) development has become a widely adopted approach that fosters collaboration among individuals from around the world on large-scale software projects hosted on platforms like GitHub. These collaborations typically take place through conversations in the context of pull requests or issues [1, 23], where developers discuss and review each other's contributions. However, as with all types of interactions, conversations in open-source development can become uncivil [9, 10]. These incivilities in GitHub conversations can manifest in various forms, including *frustration*, *sarcasm*, *insults*, and *personal attacks*, which may negatively impact developer engagement and community well-being. Such interactions can discourage contributions, reduce productivity, and create a hostile environment for newcomers. Previous studies have identified different categories of uncivil comments in GitHub conversations [7, 9, 10, 19, 24]. These range from subtle expressions of impatience or irony to explicit threats and offensive language. Understanding these behaviors is essential for designing automated moderation tools and fostering healthier online communities. Table 2 summarizes and gives examples of common types of incivilities observed in GitHub conversations.

Table 2: Type of incivility extracted from [7]

Incivility	Definition	Example
Bitter Frustration	Expressing strong frustration, displeasure, or annoyance	*Fixing clippy warnings isn't adding anything for users*
Impatience	Expressing dissatisfaction due to delays	*I am locking this thread. It is becoming useless*
Mocking	Ridiculing or making fun of someone in a disrespectful way	*congrats, you won an award for the best support of the month*
Irony	Using language to imply a meaning that is opposite to the literal meaning, often sarcastically	*Ok, you win, have fun arguing forever instead of proposing a solution*
Vulgarity	Using offensive or inappropriate language	*It honestly looks like they don't give a sh*t, rules this out as an option for me!*
Threat	Issuing a warning that implies a negative consequence	*This is the final notice. Be honest, respectable, and collaborative*
Entitlement	Expecting special treatment or privileges	*Or you could start contributing instead of bashing people who actually do the work*
Insulting	Making derogatory remarks towards another person or project	*Seems like only thing you can do so far is talk, come back when you will have any skill to show.*
Identity Attack/ Name-Calling	Making derogatory comments based on race, religion, gender, sexual orientation, or nationality	*I would not be surprised if this database is maintained by the [nationality].*

3 Study Settings

To define the goal, and research questions (RQs), we rely on the Goal-Question-Metric (GQM) template [3]. Our goal is to: **analyze** the effectiveness of SLMs; **for the purpose of** detecting incivility; **with respect to** (i) their classification performance at different levels of granularity – coarse-grained (civil vs. uncivil) and fine-grained (specific types); (ii) the impact of different prompting techniques; and (iii) their performance in comparison with traditional Machine Learning (ML) models, including those using standard NLP text encoding techniques; **from the viewpoint of** researchers; **in the context of** GitHub conversations. We detail each RQ as follows:

RQ$_1$: *To what extent can SLMs effectively detect incivility in GitHub conversations across different levels of granularity?* – Understanding the ability of SLMs to classify incivility is crucial for automated moderation. Thus, **RQ$_1$** aims to assess the effectiveness of SLMs in classifying incivility at two levels of granularity: (i) coarse-grained classification (civil vs. uncivil) and (ii) fine-grained classification, distinguishing between different types of incivility.

By examining how well SLMs identify instances of incivility, we aim to determine whether their accuracy differs based on classification complexity. To answer **RQ$_1$**, we will evaluate multiple SLMs using well-known metrics, such as precision, recall, and F1-score, on a labeled dataset of GitHub conversations.

RQ$_2$: *How do different prompting techniques impact the effectiveness of SLMs in detecting incivility in GitHub conversations?* – SLMs vary in parameter count and sensitivity to prompt strategy. In this context, **RQ$_2$** examines the influence of prompting strategies – such as *zero-shot*, *one-shot*, *few-shot*, *auto-CoT*, and *role-based* on the performance of incivility detection in different SLMs. By comparing SLMs and prompting techniques, we aim to identify which configurations yield the most accurate and reliable results for automated moderation in software development discussions. To answer **RQ$_2$**, we will measure the impact of these prompting strategies on performance scores, using existing datasets.

RQ$_3$: *How effective are SLMs compared to ML models combined with NLP text encoding techniques in detecting incivility in GitHub conversations?* – **RQ$_3$** aims to evaluate and compare the effectiveness of the best-performing SLM configuration of RQ$_2$ against five traditional ML models: *Multinomial Naive Bayes (MNB)* , *Logistic Regression Classifier (LRC)*, *Random Forest Classifier (RFC)*, *AdaBoost Classifier (ABC)*, and *DistilBERT (DBERT)* [27]. We also included two standard NLP text encoding techniques: Bag of Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) [30]. To answer **RQ$_3$**, we apply each ML model combined with the NLP techniques to the same labeled dataset and evaluate their effectiveness using precision, recall, and F1-score. This analysis helps determine if SLMs are better at detecting incivility made by developers in GitHub conversations.

3.1 Study Steps and Procedures

Step 1: Select and preprocess datasets for analysis. We use two datasets: one proposed by Rahman et al. [24] and another released by Ehsani et al. [7], to support the classification of incivility comments at two levels of granularity: fine and coarse. The dataset proposed by Rahman et al. [24] focuses on coarse-grained level, including 6.4K uncivil comments from code review discussions and their civil counterparts, all extracted from GitHub projects. This level of granularity (*uncivil* vs *civil*) provides an overview of the ability of SLMs to detect incivility directly. In contrast, the dataset by Ehsani et al. [7] captures more complex discussions by considering various types of incivility in GitHub conversations, which we regard as a fine-grained level. It comprises 404 locked conversations (issues and pull requests) on GitHub, with 5,961 comments annotated with different categories of uncivil tone-bearing discussion (see Table 2). This dataset allows us to evaluate whether models can distinguish nuanced forms of incivility expressed by developers.

After selecting the datasets, we applied the following data preprocessing steps to create two datasets with different granularities: (1) missing data removal; (2) label encoding; (3) dataset merging; (4) and duplicate removal. For the coarse-grained level, we began with the Ehsani et al. [7] dataset, removing 10 entries where the comment body was missing. We then performed label encoding, mapping the original class labels to binary categories: instances labeled as *Bitter Frustration*, *Impatience*, *Mocking*, *Vulgarity*, *Irony*, *Insulting*, *Identity Attacks*, *Entitlement* and *Threat* were classified as *Uncivil*, while *None* was categorized as *Civil*. Next, we merged this dataset with the Rahman et al. [24] dataset to create a unified dataset. Finally, we identified and removed 97 duplicate entries to ensure data integrity. After preprocessing, the final coarse-grained dataset comprised 6,879 examples, with 4,943 instances labeled as civil (71.85%) and 1,936 instances labeled as uncivil (28.15%).

For the fine-grained level, we removed the same 10 entries with missing comment bodies from the Ehsani et al. [7] dataset. We then standardized inconsistent class labels, merging *identity attack/name-calling* and *identity attacks/name-calling* into *identity attack/name-calling*. Additionally, we renamed columns ("actual" to "tbdf" and "message" to "comment_body"). We considered merging the *Civil* classes from the Rahman et al. [24] dataset with the *None* class in our dataset. However, given the substantial number of *None* instances, and to keep the *Civil* instances balanced, we opted not to proceed with the merge. Finally, we identified and removed 88 duplicate entries, resulting in a fine-grained dataset of 5,871 samples.

Table 3: Overviews of the consolidated datasets

Granularity	Type	# of comments	Pct. (%)
Coarse-grained	Civil	4,943	71.85
	Uncivil	1,936	28.15
Fine-grained	None	4,507	76.77
	Bitter Frustration	492	8.38
	Impatience	266	4.53
	Mocking	176	3.00
	Insulting	174	2.96
	Vulgarity	71	1.21
	Entitlement	70	1.19
	Irony	64	1.09
	Identity Attacks	28	0.48
	Threat	23	0.39

Step 2: Select SLM models for analysis. We selected a range of SLMs from different families, including Gemma, Llama, Mistral, DeepSeek, Phi, and GPT-based models, to evaluate their effectiveness in detecting incivility in GitHub conversations. We selected the models according to the following criteria: popularity, cost, scale, and availability. We describe these criteria as follows.

Popularity. We evaluated the popularity of the model based on its usage and validation within the research and development community, using the number of downloads of model families on Hugging Face[1] as an indicator of adoption. Our rationale was that models with high download counts are generally well-regarded and have demonstrated effectiveness across various NLP tasks. *Cost.* We considered both operational and computational costs. Closed-source models, such as GPT-4o mini, incur usage fees through cloud APIs. In contrast, open source-models, e.g., Gemma, Llama, Mistral, DeepSeek, and Phi are free, but require significant local infrastructure for deployment. Thus, we balanced cost-effectiveness with the required computational resources. *Scale.* We consider model size (parameter count) as a proxy for scale, which affects inference speed, memory requirements, and reasoning capabilities. We selected the highest configuration supported by our setup, ranging from 3 to 14 billion parameters. While OpenAI does not disclose the exact parameter count of GPT-4o mini, it is described as a small model [20], suggesting it likely has fewer than 10 billion parameters. *Availability.* We considered accessibility, where cloud-based models offer convenience at a cost, while open-source models require

[1]https://huggingface.co/models?pipeline_tag=text-generation&sort=downloads

setup but are fee-free, impacting both cost and scalability. Table 4 overviews the selected SLM models for this study.

Table 4: Characteristics of the selected SLMs

Name	Parameters (in billions)	Size (GB)	Licence Type
Gemma	7	5	Open Source
Gemma2	9	5.4	Open Source
Mistral	7	4.1	Open Source
Mistral-nemo	12	7.1	Open Source
Deepseek-R1	8	4.9	Open Source
Deepseek-R1	14	9	Open Source
LLaMA3.1	8	4.9	Open Source
LLaMA3.2	3	2	Open Source
Phi4	14	9.1	Open Source
GPT-4o mini	-	-	Closed Source

Step 3: Design prompts to evaluate the effectiveness of different SLMs. We carefully designed prompts to ensure consistency, fairness, and reproducibility across models. We used four distinct prompting strategies: *zero-shot*, *one-shot*, *few-shot*, *auto-CoT*, and *role-based* prompting (see Section 2). We selected these strategies for their simplicity, making them easy to implement in practical scenarios, and their frequent use in prior studies [2, 4, 16, 22, 35]. To systematically test multiple prompting strategies, we developed a script-based prompt factory that dynamically injected data from our consolidated datasets into predefined templates, ensuring consistency and efficiency in prompt generation.

Each prompt template instructed the model to classify a given text according to the classification task. At a coarse-grained level, the model only needed to distinguish between *civil* and *uncivil* comments without further differentiation, simplifying the task. At the fine-grained level, the prompts further specified the types of incivility, such as *Bitter Frustration*, *Mocking*, *Impatience*, *Irony*, *Vulgarity*, *Insulting*, *Threat*, *Identify Attacks*, *Entitlement*, and *None*. This allowed the model to make more detailed distinctions within the uncivil category. We structured each prompt using predefined placeholders to ensure clarity and standardization. These placeholders represent key elements such as the classification task. Table 5 outlines the placeholders used in our prompt templates, distinguishing between fine-grained and coarse-grained classification.

Table 6 provides an overview of the prompt templates for four prompt strategies and three combinations used in the fine-grained and coarse-grained classification of incivility. Each strategy differs in complexity and the degree of guidance given to the model.

Table 6: Overview of prompt templates by strategy

Strategy	Prompt Structure
Zero-shot	Consider the following message: `{input_text}`. Your task is `{task}`. `{answer_template}`

One-shot	Your task is `{task}`. `{answer_template}`.
	Consider the following example messages:
	`{list_of_examples}`
	Now, analyze the following message:
	`{input_text}`

Few-shot	Your task is `{task}`. `{answer_template}`.
	Consider the following example messages:
	`{list_of_examples}`
	Now, analyze the following message:
	`{input_text}`

Auto-CoT	Consider the following message: `{input_text}`. Your task is `{task}`. `{auto_CoT_promotion}`. `{answer_template_auto_CoT}`
Role-based	`{model_instruction}`. `{task_instruction}`. Consider the following message: `{input_text}`. Your task is `{task}`. `{answer_template}`

In summary, the zero-shot prompt has the simplest structure, containing only the task specification, input text, and answer template, without any context or examples. The one-shot prompt includes a single labeled example per class (randomly selected from the dataset) to guide classification, following the format of the `{list_of_examples}` placeholder to help the model maintain a consistent pattern. The one-shot approach was applied to both levels of granularity: at the fine-grained level, it presents one example per type of incivility, and the None case, totaling 10 examples; at the coarse-grained level, it includes one example for civil and uncivil comments, totaling two examples, along with classification criteria.

The few-shot prompt follows a similar structure but varies by granularity: at the fine-grained level, it presents three examples per type of incivility ($k = 3$, randomly selected), totaling 30 examples; at the coarse-grained level, it includes three examples (k=3, randomly selected) for civil and uncivil comments, totaling six examples, along with classification criteria. This approach mirrors the example structure of the one-shot prompt. In auto-CoT prompts, additional instructions encourage the model to reason through its response in structured steps before providing the final answer, promoting a reasoning-driven approach. Finally, in role-based prompts, the model receives a detailed description of its role and the task to be performed, ensuring more context-aware responses.

Step 4: Model execution and experiment setup. To evaluate and compare different SLMs, we deployed models using two providers: OpenAI API [2] (cloud-based) and Ollama [3](local). For local execution, we used the Ollama tool to run multiple models from different families of SLMs: Meta (Llama 3.1, and Llama 3.2), Mistral AI (Mistral, and Mistral-Nemo), Google (Gemma and Gemma 2), Deepseek (Deepseek-R1) and Microsoft (Phi 4). In the cloud-based setup, we deployed GPT-4o mini via OpenAI API, leveraging its optimized infrastructure for scalability and performance. The local models were executed on a machine with the following specifications: Processor (AMD Ryzen 7 7700, 8 cores, 16 threads, 3.8-5.3GHz, 40MB cache), Memory (64GB DDR5 5200MHz), and GPU (Gigabyte RTX 4070 Ti Super 16GB VRAM). To ensure consistency in the model outputs, we set the temperature to 0 for all models. We did this to produce fully deterministic conclusions, by eliminating stochastic variance that could obscure the isolated effect of different request of prompting strategies in our batch inference setups. Moreover, for classification tasks, the zero value improves label consistency and reduces hallucinatory rational text, thereby increasing the reproducibility and fidelity of downstream evaluation [25].

Step 5: Select, build and train ML models. We consider five ML models to detect incivility in GitHub conversations: *Multinomial Naive Bayes (MNB)*, *Logistic Regression Classifier (LRC)*, *Random Forest Classifier (RFC)*, *AdaBoost Classifier (ABC)*, and *DistilBERT (DBERT)* [27], a light pre-trained version of BERT, the transformer-based model proposed by Google. The selection of MNB, LRC, RFC, and DBERT was based on a prior work on incivility detection [11]. In contrast, the inclusion of the ABC model represents an intentional extension of the baseline set, providing an additional comparative perspective that complements and enriches our analysis. The machine learning models were implemented using the `sklearn`

[2]https://openai.com/index/openai-api
[3]https://ollama.com

Table 5: Placeholders used in our prompt templates

Placeholder	Fine-grained Classification	Coarse-grained Classification
`{task}`	to classify the message into one of the specified classes: Bitter Frustration, Impatience, Mocking, Irony, Vulgarity, Threat, Identify Attack/Name Calling, Entitlement, Insulting or None.	to determine if the message is **CIVIL** or **UNCIVIL**.
`{input_text}`	The text to be classified.	The text to be classified.

`{list_of_examples}`	Example n: {example_label_n}
	Response for Example n: {"label":"label_n"}
		Example n: {example_label_n}
		Response for Example n: {"label":"label_n"}

`{answer_template}`	Return the result as a JSON with the following format: {{"label": "Bitter Frustration" OR "Impatience" OR "Mocking" OR "Irony" OR "Vulgarity" OR "Identify Attacks/Name Calling" OR "Threat" OR "Entitlement" OR "Insulting" OR "None"}}	Return the result as JSON with the following format: {{"label": "CIVIL" OR "UNCIVIL"}}
`{model_instruction}`	**You are a GitHub moderator of conversations** that classifies the tone of messages in GitHub discussions as "Bitter Frustration" or "Impatience" or "Mocking" or "Irony" or "Vulgarity" or "Identify Attacks/Name Calling" or "Threat" or "Entitlement" or "Insulting" or "None"	**You are a GitHub moderator of conversations** that classifies the tone of messages in GitHub discussions as "CIVIL" or "UNCIVIL"
`{task_instruction}`	Mocking are those messages that involves ridiculing or making fun of someone in a disrespectful way; Identity Attack/Name Calling are those messages that involves making derogatory comments based on race, religion, gender, sexual orientation, or nationality; Bitter Frustration are those messages that involves strong frustration, displeasure, or annoyance; Impatience are those messages that involves conveys dissatisfaction due to delays; Threat are those messages that involves issuing a warning that implies a negative consequence; Irony are those messages that involves uses language to imply a meaning opposite to the literal one, often sarcastically; Vulgarity are those messages that involves offensive or inappropriate language; None are messages that do not contain any type of aforementioned incivility; Entitlement are those messages that involves expecting special treatment or privileges; and Insulting are those messages that involves making derogatory remarks towards another person or project; and None are messages that do not contain any type of aforementioned incivility.	CIVIL messages are those that are respectful, constructive, and polite. UNCIVIL messages include any of the following tones: Mocking which involves ridiculing or making fun of someone in a disrespectful way; Identity Attack or name-calling which consists of making derogatory comments based on race, religion, gender, sexual orientation, or nationality; Bitter Frustration which expresses strong frustration, displeasure, or annoyance; Impatience which conveys dissatisfaction due to delays; Threat which involves issuing a warning that implies a negative consequence; Irony which uses language to imply a meaning opposite to the literal one, often sarcastically; Vulgarity which includes offensive or inappropriate language; Entitlement which reflects an expectation of special treatment or privileges; and Insulting which involves making derogatory remarks towards another person or project
`{auto_CoT_promotion}`	Let's think step by step. Provide reasoning before giving the response	Let's think step by step. Provide reasoning before giving the response
`{answer_template_auto_CoT}`	Your response should be only a json in the following format: {{"label": "Bitter Frustration" OR "Impatience" OR "Mocking" OR "Irony" OR "Vulgarity" OR "Identify Attacks/Name Calling" OR "Threat" OR "Entitlement" OR "Insulting" OR "None", "reasoning": "Your reasoning here"}}	Your response should be only a json in the following format: {{"label": "CIVIL" or "UNCIVIL", "reasoning": "Your reasoning here"}}

Python library with default hyperparameters. For the fine-grained classification task, Logistic Regression was employed using the One-vs-Rest (OvR) strategy, in which a separate binary classifier is trained for each class against all others.

We also used two different NLP text encoding techniques - Bag of Words (BoW), which converts the document corpus into an array of text token/word count and Term Frequency-Inverse Document Frequency (TF-IDF), which weights the relevance of each token/word in the document [30]. For the DBERT, we use the raw textual comments as input, without additional preprocessing, to fine-tune a classification layer. We train the models using a 10-fold cross-validation strategy [15]. This strategy randomly partitions the dataset into 10 folds of equal size, each with the same proportion of the various criticality classes. A single fold is then used as a test set, while the remaining ones are employed or training the model, i.e., they are independent of each other. The final performance for each model is obtained by averaging the results across the folds. Moreover, to maintain a degree of comparability, we did not apply hyperparameter tuning to the ML models, given that the SLMs were also used without fine-tuning, or data balancing, as it could result in excessive generation of synthetic data or the loss of real data. However, we acknowledge that this does not fully eliminate potential disparities, especially considering the limited transparency regarding the SLMs' original training conditions.

Step 6: Evaluation of effectiveness with traditional metrics. To evaluate the effectiveness of each SLM and ML model, we analyzed confusion matrices and their derived metrics, enabling a comparative evaluation of different levels of granularity, prompts, SLMs, and ML models used in this study. The confusion matrices provided insight into the model predictions by categorizing the output into true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). More specifically, we used the following well-known metrics [8]: (i) *Precision* measures the proportion of true positive observations among all the predicted positive observations ($Pr = \frac{TP}{TP+FP}$). A high precision indicates that the model produces fewer false positives, which is particularly important in applications where false positives have significant consequences; (ii) *Recall* represents the proportion of true positive observations out of all actual positive observations that were correctly identified by the model ($Re = \frac{TP}{TP+FN}$). A high recall indicates that the model captures most of the positive cases, reducing the number of false negatives; and (iii) *F1-score* (F1) is the harmonic mean of precision and recall, providing a balanced measure when there is an uneven class distribution. It is defined as: ($F1 = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$). This metric is useful when both false positives and false negatives are critical, as it balances the trade-off between precision and recall.

4 Results and Discussion

4.1 Effectiveness of SLMs (RQ$_1$)

To answer **RQ$_1$**, we analyzed the effectiveness of the SLMs across five prompting strategies: *zero-shot*, *one-shot*, *few-shot*, *auto-CoT*, and *role-based.* Table 7 lists the average performance scores for each SLM at the coarse-grained level (civil vs. uncivil).

Table 7: Average performance scores of SLMs at the coarse-grained level across strategies

Model	Civil			Uncivil		
	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.82	0.94	0.87	0.75	0.46	0.57
deepseek-r1:14b	0.77	0.98	0.86	0.84	0.24	0.35
mistral-nemo:12b	0.83	0.85	0.84	0.62	0.56	0.58
gemma2:9b	0.83	0.90	0.86	0.68	0.52	0.59
llama3.1:8b	0.81	0.93	0.86	0.70	0.44	0.54
deepseek-r1:8b	0.75	0.98	0.85	0.79	0.15	0.24
gemma:7b	0.83	0.68	0.74	0.47	0.64	0.53
mistral:7b	0.78	0.96	0.86	0.77	0.32	0.44
llama3.2:3b	0.84	0.73	0.77	0.51	0.63	0.54
gpt-4o-mini	0.84	0.93	0.88	0.75	0.53	0.62

The effectiveness of SLMs at the coarse-grained level across different prompt strategies. Table 7 shows that the performance of SLMs varies depending on the target class and model's size. In general, most SLMs achieve a strong performance in detecting civil comments, with a high average F1 (0.74-0.88), with models such as *gpt-4o-mini*, *phi4:14b*, and *mistral-nemo:12b* reaching up to 0.84. Even smaller models like *gemma:7b*, *mistral:7b*, and *llama3.2:3b* achieve solid average F1-scores at least 0.74 for civil comments. However, their performance drops notably when handling uncivil comments. Only *gpt-4o-mini* is outperforming on average with F1 of 0.60 for this category, while models such as *gemma:7b*, *gemma2:9b*, *mistral:7b*, *phi4:14b*, *mistral-nemo:12b*, *llama3.2:3b*, and *llama3.1:8b* range from 0.40-0.59. Interestingly, *deepseek-r1:8b* and *deepseek-r1:14b* are both below 0.40.

In terms of precision, the results are particularly encouraging for civil comments, where most SLM models consistently achieve high average precision scores (0.75-0.84). This demonstrates their strong ability to minimize false positives and accurately identify non-offensive content. Notably, even in the more challenging task of detecting uncivil comments, several models — including *deepseek-r1:14b*, *deepseek-r1:8b*, and *mistral:7b* — exhibit impressive performance, attaining high average precision scores (0.77-0.84). These results indicate that, among the instances classified as uncivil by these models, a large proportion are indeed truly uncivil, reinforcing their reliability in detecting harmful content with low rates of misclassification. In contrast, the *gemma:7b*, *llama3.2:3b* present scores around (0.47-0.51), indicating a high rate of FP predictions for uncivil comments. In terms of recall, a similar trend is observed, with most SLMs achieving high average scores around 0.90 to *phi4:14b*, *deepseek-r1:14b*, *deepseek-r1:8b*, *gpt-4o-mini*, *mistral:7b*, *llama3.1:8b*, and *gemma2:9b*) for civil comments. For uncivil comments, most SLMs exhibit limited performance, with only *gemma:7b* and *llama3.2:3b* outperforming 0.60. Notably, *deepseek-r1:8b* shows a particularly low average recall, below 0.20.

Furthermore, we highlight that *gemma2:9b*, *mistral-nemo:12b*, and *gpt-4o-mini* consistently demonstrated strong performance across all three evaluation metrics and both target classes. This observation reinforces the tendency that SLMs with higher parameter counts (9-14b) generally exhibit superior overall effectiveness, suggesting that standard prompting strategies are particularly well-suited for such models. To investigate whether there are statistically significant differences in performance among the evaluated SLMs, we conducted the Friedman non-parametric test [12] with Nemenyi's post-hoc pairwise comparisons, using a significance level of $alpha = 0.05$. In coarse-grained classification, statistically significant results were observed primarily with GPT models for F1, and with Mistral-Nemo for Precision.

Finding 1: SLMs effectively detect civil comments, with high Pr, Re, and F1 across all sizes. In contrast, uncivil comments detection is more challenging and highly dependent on size. For real-world applications like moderating uncivil comments in GitHub conversations, the top SLMs with 9B+ parameters, such as *gemma2:9b*, *mistral-nemo:12b*, and *gpt-4o-mini* are more suitable for production use.

The effectiveness of SLMs at the fine-grained level across different prompt strategies. Table 8 overviews the average performance scores for each SLM in detecting different types of incivility (fine-grained level) across prompt strategies.

Table 8: Average performance scores of SLMs at the fine-grained level across strategies

Model	Bitter Frustration			Impatience			Vulgarity			Irony			Identity Attack		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
deepseek-r1:14b	0.17	0.44	0.23	0.18	0.15	0.12	0.36	0.11	0.15	0.01	0.00	0.00	0.16	0.04	0.05
phi4:14b	0.23	0.34	0.25	0.12	0.30	0.16	0.42	0.21	0.26	0.05	0.10	0.06	0.13	0.19	0.11
mistral-nemo:12b	0.16	0.47	0.21	0.12	0.17	0.09	0.40	0.18	0.24	0.04	0.06	0.05	0.17	0.08	0.06
gemma2:9b	0.17	0.47	0.24	0.06	0.28	0.09	0.30	0.08	0.11	0.03	0.05	0.04	0.12	0.14	0.12
llama3.1:8b	0.14	0.39	0.19	0.08	0.30	0.11	0.29	0.17	0.14	0.06	0.10	0.07	0.20	0.11	0.10
deepseek-r1:8b	0.18	0.28	0.20	0.09	0.28	0.10	0.22	0.11	0.10	0.04	0.05	0.04	0.12	0.09	0.08
gemma:7b	0.15	0.31	0.16	0.06	0.27	0.08	0.25	0.19	0.20	0.02	0.18	0.02	0.05	0.11	0.05
mistral:7b	0.12	0.50	0.15	0.07	0.19	0.09	0.28	0.09	0.11	0.04	0.04	0.03	0.06	0.12	0.07
llama3.2:3b	0.16	0.27	0.16	0.06	0.24	0.08	0.22	0.14	0.07	0.01	0.17	0.02	0.05	0.06	0.04
gpt-4o-mini	0.23	0.52	0.31	0.15	0.36	0.20	0.51	0.30	0.37	0.07	0.14	0.09	0.21	0.18	0.18
Average	0.17	0.40	0.21	0.10	0.25	0.11	0.32	0.16	0.18	0.04	0.09	0.04	0.13	0.11	0.09

Model	Threat			Insulting			Entitlement			Mocking			None		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
deepseek-r1:14b	0.36	0.37	0.32	0.27	0.15	0.05	0.01	0.01	0.01	0.15	0.18	0.16	0.88	0.77	0.82
phi4:14b	0.41	0.38	0.34	0.38	0.20	0.08	0.08	0.04	0.03	0.14	0.31	0.19	0.92	0.74	0.82
mistral-nemo:12b	0.43	0.24	0.24	0.45	0.17	0.06	0.09	0.11	0.07	0.16	0.25	0.19	0.93	0.59	0.68
gemma2:9b	0.28	0.40	0.32	0.26	0.23	0.11	0.13	0.07	0.06	0.17	0.29	0.21	0.93	0.54	0.67
llama3.1:8b	0.36	0.24	0.24	0.06	0.20	0.04	0.06	0.11	0.05	0.14	0.26	0.16	0.94	0.44	0.58
deepseek-r1:8b	0.52	0.18	0.19	0.14	0.16	0.06	0.08	0.04	0.03	0.09	0.17	0.12	0.87	0.64	0.70
gemma:7b	0.11	0.45	0.17	0.18	0.17	0.06	0.06	0.03	0.03	0.09	0.33	0.13	0.89	0.10	0.17
mistral:7b	0.12	0.16	0.05	0.10	0.22	0.08	0.09	0.05	0.04	0.19	0.10	0.12	0.90	0.43	0.49
llama3.2:3b	0.34	0.17	0.11	0.01	0.20	0.02	0.27	0.05	0.06	0.08	0.38	0.12	0.94	0.19	0.29
gpt-4o-mini	0.51	0.54	0.51	0.34	0.22	0.26	0.13	0.10	0.10	0.25	0.29	0.27	0.94	0.69	0.79
Average	0.34	0.31	0.25	0.22	0.19	0.08	0.10	0.06	0.05	0.15	0.26	0.17	0.91	0.51	0.60

In general, SLMs have limited performance in handling different types of incivilities, with Pr and F1 values predominantly below 0.30. Some models, such as *phi4:14b*, *deepseek-r1:14b*, *mistral-nemo:12b*, and *gpt-4o-mini*, tend to be more consistent in both Pr and F1 values. However, some interesting patterns emerge when analyzing individual incivility types. *Bitter Frustration* shows the greatest variation among models, e.g., *gpt-4o-mini* achieves a F1 of 0.31, other models perform significantly worse, indicating that this type poses semantic challenges. In contrast, *Threat* tends to yield better overall results, especially in Pr, suggesting that models are more adept at detecting this more explicit form of incivility. Conversely, types such as *Irony*, *Impatience*, *Insulting*, *Entitlement*, and *Mocking* remain consistently more difficult to detect, with Pr and F1 values often below or around 0.27. These types appear to depend on subtle linguistic cues and contextual interpretation, requiring more advanced reasoning capabilities from the models. We apply the same statistical analysis as in coarse-grained to compare model performance. Unlike the coarse-grained results, no statistically significant differences were found among the evaluated models across most specific incivility categories.

Finding 2: SLMs still present a limited performance in detecting incivility, particularly struggling with implicit content such as *Irony*, *Impatience*, *Insulting*, *Entitlement*, and *Mocking*. This suggests the need for more structured prompting strategies to enhance the sensitivity and coverage of SLMs in content moderation tasks.

4.2 Effect of Different Prompt Strategies (RQ$_2$)

To answer **RQ$_2$**, we analyze how five prompting strategies - *zero-shot*, *one-shot*, *few-shot*, *auto-CoT*, and *role-based* - influence SLMs' performance in detecting incivilities at both coarse- and fine-grained levels. Tables 9 and 10 show the performance scores of each prompt strategy at the coarse- and fine-grained level, grouped by SLM, respectively. Furthermore, cells highlighted in a Green color indicate models that achieved the highest precision (Pr) and F1-score (F1) by prompt strategy. Finally, those in a Red color correspond to models with the lowest Pr and F1.

Table 9: Overviews of performance variation of prompt strategies at the coarse-grained level grouped by SLMs

Model	Civil														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.81	0.95	0.87	0.82	0.94	0.87	0.82	0.93	0.87	0.81	0.95	0.87	0.83	0.92	0.87
deepseek-r1:14b	0.78	0.97	0.87	0.74	1.00	0.85	0.74	1.00	0.85	0.77	0.98	0.86	0.82	0.95	0.88
mistral-nemo:12b	0.82	0.92	0.87	0.85	0.78	0.81	0.84	0.81	0.82	0.81	0.93	0.87	0.86	0.84	0.85
gemma2:9b	0.83	0.88	0.86	0.83	0.91	0.86	0.82	0.92	0.87	0.83	0.89	0.86	0.82	0.93	0.87
llama3.1:8b	0.82	0.90	0.86	0.80	0.93	0.86	0.79	0.95	0.86	0.80	0.94	0.87	0.84	0.91	0.87
deepseek-r1:8b	0.73	1.00	0.84	0.74	0.99	0.85	0.76	0.94	0.84	0.77	0.96	0.86	0.74	1.00	0.85
gemma:7b	0.81	0.71	0.76	0.84	0.55	0.66	0.84	0.59	0.69	0.84	0.65	0.73	0.82	0.90	0.86
mistral:7b	0.76	0.98	0.86	0.81	0.93	0.87	0.81	0.93	0.87	0.77	0.98	0.86	0.78	0.97	0.86
llama3.2:3b	0.84	0.64	0.73	0.85	0.74	0.79	0.79	0.93	0.85	0.86	0.56	0.68	0.86	0.78	0.82
gpt-4o-mini	0.83	0.95	0.88	0.83	0.92	0.88	0.83	0.93	0.88	0.83	0.93	0.88	0.85	0.92	0.89
Average	0.80	0.89	0.84	0.81	0.87	0.83	0.80	0.89	0.84	0.81	0.88	0.83	0.82	0.91	0.86

Model	Uncivil														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.77	0.41	0.54	0.74	0.47	0.58	0.73	0.49	0.59	0.77	0.42	0.54	0.71	0.53	0.61
deepseek-r1:14b	0.82	0.30	0.44	0.89	0.09	0.16	0.89	0.09	0.16	0.84	0.26	0.39	0.78	0.46	0.58
mistral-nemo:12b	0.71	0.48	0.57	0.53	0.64	0.58	0.55	0.60	0.57	0.71	0.44	0.55	0.61	0.65	0.63
gemma2:9b	0.64	0.55	0.59	0.68	0.51	0.59	0.70	0.50	0.58	0.66	0.55	0.60	0.72	0.47	0.57
llama3.1:8b	0.66	0.51	0.58	0.70	0.40	0.51	0.72	0.35	0.47	0.73	0.40	0.52	0.70	0.56	0.62
deepseek-r1:8b	0.91	0.04	0.07	0.79	0.11	0.19	0.60	0.22	0.32	0.74	0.28	0.40	0.90	0.11	0.20
gemma:7b	0.44	0.59	0.50	0.39	0.74	0.51	0.41	0.71	0.52	0.43	0.67	0.53	0.67	0.50	0.57
mistral:7b	0.83	0.20	0.32	0.72	0.43	0.54	0.71	0.45	0.55	0.81	0.24	0.37	0.81	0.28	0.42
llama3.2:3b	0.43	0.69	0.53	0.50	0.66	0.57	0.66	0.36	0.46	0.40	0.77	0.53	0.54	0.67	0.60
gpt-4o-mini	0.78	0.49	0.60	0.73	0.53	0.61	0.74	0.53	0.62	0.75	0.51	0.61	0.74	0.60	0.66
Average	0.70	0.43	0.47	0.67	0.46	0.48	0.67	0.43	0.48	0.69	0.45	0.50	0.72	0.48	0.55

Performance variation of prompting strategies in detecting civil and uncivil comments (coarse-grained level). When comparing the detection of civil and uncivil comments, we observe that for civil comments, most prompt strategies, regardless of the SLM used, achieve high performance scores for Pr, Re, and F1, around 0.73-0.86, 0.55-1.00, and 0.66-0.89, respectively. In contrast, for uncivil comments, there is a significant variation in performance scores across strategies. Strategies like *role-based* tend to improve Pr, Re, and F1 for civil comments compared to *zero-shot*, with average increases of 0.02, 0.03, and 0.02, respectively, across different SLMs. For uncivil comments, the improvements are even more evident, with average increases of 0.03 in Pr, 0.05 in Re, and 0.07 in F1, suggesting that this strategy is particularly beneficial for enhancing the detection of problematic comments. However, strategies such as *few-shot* often yield stagnant performance. In the case of civil comments, this approach showed no improvement across Pr, Re, or F1. For uncivil comments, Pr decreases on average by 0.03, while Re remains unchanged (0.00) and F1 shows only a slight improvement of 0.01, indicating a limited ability of this strategy to support the detection of offensive or inappropriate comments.

When we look at specific prompt strategies, we observe that *zero-shot* achieves strong performance for civil comments, with Pr (0.73-0.84), Re (0.64-1.00), and F1 (0.73-0.88) across SLMs. However, for uncivil comments, its performance dropped in terms of Pr (0.43-0.91), Re (0.04-0.69), and F1 (0.07-0.60). On average, zero-shot results for uncivil comments across SLMs show Pr, Re, and F1 of 0.70, 0.43, and 0.47, respectively. These results suggest that, in the absence of examples, SLMs exhibit a default bias toward classifying comments as civil. In contrast, adding a single example (*one-shot*) maintains strong performance for civil comments with Pr (0.74-0.85), Re (0.55-1.00), and F1 (0.66-0.88), while offering slight improvements for uncivil comments compared to *one-shot* for Re and F1. In the one-shot prompt, the detection of uncivil comments is achieved with Pr (0.39-0.89), Re (0.09-0.74), and F1 (0.16-0.61), with average values across SLMs of 0.67, 0.43, 0.48, respectively.

In the case of inclusion of multiple examples (*few-shot*), it appears not to significantly enhance SLMs' scores about *one-shot*. When averaging across all SLMs for the uncivil class, the model's performance remains stable, with Pr (0.60), Re (0.43), and F1 (0.48). These results suggest that adding additional examples does not necessarily improve the performance, and that a single well-chosen example may be sufficient to guide the model's behavior effectively.

The *auto-CoT* strategy obtained a robust performance, in which civil comments remained consistently high with Pr (0.77-0.86), Re (0.56-0.98), and F1 (0.68-0.88). For uncivil comments, performance varied significantly, with Pr (0.40-0.84), Re (0.24-0.77), and F1 (0.37-0.61), with an average of 0.69, 0.45, 0.50, respectively, across SLMs. As expected, this suggests that the structured reasoning encouraged by Chain-of-Thought helps models make more informed and conservative decisions, especially when handling uncivil comments.

Finally, *role-based prompting* obtained some of the best results, outperforming *few-shot* and *auto-CoT* strategies for civil comments, with Pr (0.74-0.86), Re (0.78-1.00), and F1 (0.82-0.89). In contrast, the performance in uncivil comments was more moderate, with Pr (0.54-0.90), Re (0.11-0.67), and F1 (0.20-0.66), with average scores of 0.72 (Pr), 0.48 (Re), and 0.55 (F1) across SLMs. These results suggest that adopting a responsible role (e.g., moderator) enhances the SLMs' ability to generalize moderation criteria more effectively and apply them more consistently. To assess the impact of prompting strategies on model performance, we applied the same tests as RQ1. In the coarse-grained classification, no significant differences were observed between the strategies in terms of F1 or Precision.

Finding 3: Prompt strategies are effective for detecting civil comments, but inconsistent for uncivil ones. *Zero-shot*, *one-shot*, and *few-shot* prompts are quite limited, while structured strategies like *auto-CoT* and *role-based* prompting outperform the uncivil detection, highlighting the importance of reasoning and role awareness to improve model sensitivity.

Performance variation of prompting strategies in detecting different types of incivility (fine-grained level). Table 10 shows that, in general, all strategies resulted in varying performance scores for Pr, Re and F1, around 0.00-1, 0.00-0.96, and 0.00-0.61, respectively. For instance, the *few-shot* strategy with $k = 3$, our case, obtained poor average results across all SLMs, metrics and types of incivility with Pr (0.01-0.11) and F1 (0.02-0.15). In contrast, the *role-based prompting* technique stands out, demonstrating substantially better performance compared to the other prompt strategies in terms of Pr (0.05-0.61) and F1 (0.06-0.29).

Additionally, although Recall (Re) was not used as a primary evaluation criterion, we observed that the frequent and considerable disparities between precision and recall scores suggest that the models tend to adopt a conservative stance when analyzing the nuances of incivilities. Finally, we emphasized that the "None" class, i.e., which represents the absence of incivility, was not included in the Table 10. However, this class achieved the best performance scores, with Pr and F1 reaching 0.90 and 0.87, respectively, when using the *role-based prompting* strategy. In the fine-grained granularity, the statistical tests showed that role-based prompting outperformed

Table 10: SLMs performance scores at the fine-grained level grouped by prompt strategy

Model	Bitter Frustration														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.24	0.48	0.32	0.28	0.25	0.27	0.17	0.00	0.01	0.26	0.46	0.33	0.23	0.53	0.32
deepseek-r1:14b	0.19	0.62	0.29	0.28	0.20	0.23	0.00	0.00	0.00	0.16	0.74	0.26	0.24	0.61	0.34
mistral-nemo:12b	0.14	0.71	0.23	0.24	0.31	0.27	0.08	0.00	0.00	0.13	0.67	0.22	0.21	0.66	0.32
gemma2:9b	0.19	0.69	0.29	0.28	0.31	0.29	0.00	0.00	0.00	0.20	0.64	0.30	0.18	0.69	0.29
llama3.1:8b	0.16	0.63	0.26	0.26	0.17	0.21	0.00	0.00	0.00	0.14	0.60	0.23	0.16	0.54	0.24
deepseek-r1:8b	0.14	0.47	0.22	0.24	0.13	0.17	0.00	0.00	0.00	0.24	0.36	0.29	0.26	0.42	0.32
gemma:7b	0.13	0.59	0.21	0.25	0.05	0.09	0.00	0.00	0.00	0.16	0.48	0.24	0.19	0.43	0.26
mistral:7b	0.10	0.94	0.18	0.15	0.03	0.05	0.08	0.02	0.03	0.11	0.81	0.20	0.17	0.69	0.28
llama3.2:3b	0.19	0.40	0.26	0.32	0.08	0.12	0.00	0.00	0.00	0.15	0.50	0.23	0.13	0.36	0.20
gpt-4o-mini	0.19	0.59	0.28	0.25	0.41	0.31	0.30	0.36	0.33	0.20	0.53	0.29	0.20	0.72	0.32
Average	0.17	0.61	0.25	0.25	0.19	0.20	0.06	0.04	0.04	0.18	0.58	0.26	0.20	0.57	0.29

Model	Impatience														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.12	0.37	0.18	0.19	0.29	0.23	0.00	0.00	0.00	0.14	0.40	0.20	0.13	0.43	0.21
deepseek-r1:14b	0.12	0.29	0.17	0.18	0.38	0.24	0.00	0.00	0.00	0.21	0.03	0.06	0.41	0.06	0.11
mistral-nemo:12b	0.06	0.26	0.09	0.38	0.09	0.14	0.00	0.00	0.00	0.05	0.30	0.08	0.14	0.20	0.16
gemma2:9b	0.05	0.33	0.08	0.10	0.41	0.16	0.00	0.00	0.00	0.05	0.36	0.09	0.08	0.28	0.13
llama3.1:8b	0.06	0.49	0.11	0.21	0.18	0.20	0.00	0.00	0.00	0.07	0.49	0.12	0.07	0.32	0.11
deepseek-r1:8b	0.06	0.42	0.10	0.20	0.11	0.14	0.00	0.00	0.00	0.05	0.69	0.10	0.16	0.19	0.17
gemma:7b	0.04	0.28	0.07	0.06	0.49	0.10	0.08	0.08	0.08	0.04	0.20	0.06	0.07	0.30	0.11
mistral:7b	0.03	0.09	0.04	0.23	0.24	0.24	0.00	0.00	0.00	0.03	0.26	0.06	0.08	0.34	0.13
llama3.2:3b	0.04	0.40	0.08	0.15	0.20	0.17	0.00	0.00	0.00	0.05	0.47	0.09	0.04	0.14	0.07
gpt-4o-mini	0.10	0.42	0.16	0.18	0.34	0.24	0.19	0.33	0.24	0.11	0.47	0.18	0.18	0.21	0.19
Average	0.07	0.34	0.11	0.19	0.27	0.19	0.03	0.04	0.03	0.08	0.37	0.10	0.14	0.25	0.14

Model	Vulgarity														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.60	0.21	0.31	0.29	0.35	0.32	0.00	0.00	0.00	0.63	0.24	0.35	0.59	0.23	0.33
deepseek-r1:14b	0.48	0.14	0.22	0.25	0.25	0.25	0.00	0.00	0.00	0.43	0.04	0.08	0.67	0.11	0.19
mistral-nemo:12b	0.56	0.21	0.31	0.33	0.34	0.34	0.00	0.00	0.00	0.61	0.15	0.25	0.52	0.21	0.30
gemma2:9b	0.40	0.08	0.14	0.23	0.20	0.21	0.00	0.00	0.00	0.33	0.04	0.08	0.56	0.07	0.13
llama3.1:8b	0.54	0.10	0.17	0.14	0.54	0.22	0.00	0.00	0.00	0.50	0.10	0.16	0.29	0.11	0.16
deepseek-r1:8b	0.00	0.00	0.00	0.19	0.37	0.25	0.00	0.00	0.00	0.33	0.01	0.03	0.55	0.15	0.24
gemma:7b	0.36	0.28	0.31	0.09	0.17	0.12	0.00	0.00	0.00	0.43	0.13	0.20	0.37	0.37	0.37
mistral:7b	0.67	0.03	0.05	0.20	0.23	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.52	0.18	0.27
llama3.2:3b	0.22	0.03	0.05	0.07	0.56	0.13	0.00	0.00	0.00	0.40	0.08	0.14	0.40	0.03	0.05
gpt-4o-mini	0.52	0.24	0.33	0.51	0.34	0.41	0.53	0.41	0.46	0.50	0.22	0.31	0.51	0.27	0.35
Average	0.43	0.13	0.19	0.23	0.33	0.25	0.05	0.04	0.05	0.42	0.10	0.16	0.50	0.17	0.24

Model	Irony														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.06	0.06	0.06	0.05	0.16	0.07	0.00	0.00	0.00	0.08	0.22	0.12	0.07	0.08	0.07
deepseek-r1:14b	0.00	0.00	0.00	0.06	0.02	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mistral-nemo:12b	0.04	0.05	0.04	0.07	0.05	0.06	0.00	0.00	0.00	0.02	0.05	0.03	0.07	0.16	0.10
gemma2:9b	0.02	0.02	0.02	0.03	0.09	0.05	0.02	0.02	0.02	0.03	0.06	0.04	0.04	0.08	0.05
llama3.1:8b	0.07	0.02	0.03	0.08	0.19	0.11	0.00	0.00	0.00	0.12	0.09	0.11	0.05	0.22	0.09
deepseek-r1:8b	0.00	0.00	0.00	0.08	0.13	0.10	0.00	0.00	0.00	0.02	0.03	0.02	0.11	0.08	0.09
gemma:7b	0.01	0.08	0.02	0.02	0.13	0.03	0.03	0.02	0.02	0.01	0.19	0.02	0.01	0.48	0.02
mistral:7b	0.02	0.05	0.03	0.05	0.02	0.02	0.00	0.00	0.00	0.04	0.06	0.05	0.08	0.06	0.07
llama3.2:3b	0.01	0.06	0.02	0.01	0.44	0.02	0.00	0.00	0.00	0.02	0.30	0.04	0.01	0.05	0.01
gpt-4o-mini	0.05	0.11	0.07	0.08	0.16	0.10	0.09	0.19	0.12	0.04	0.10	0.06	0.08	0.14	0.10
Average	0.03	0.04	0.03	0.05	0.14	0.06	0.01	0.02	0.02	0.04	0.11	0.05	0.05	0.13	0.06

Model	Identify Attack														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.10	0.11	0.11	0.04	0.54	0.07	0.00	0.00	0.00	0.10	0.14	0.12	0.38	0.18	0.24
deepseek-r1:14b	0.33	0.04	0.06	0.14	0.14	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.04	0.06
mistral-nemo:12b	0.25	0.04	0.06	0.02	0.21	0.03	0.00	0.00	0.00	0.10	0.04	0.05	0.50	0.11	0.18
gemma2:9b	0.14	0.18	0.16	0.09	0.14	0.11	0.00	0.00	0.00	0.10	0.21	0.13	0.27	0.14	0.19
llama3.1:8b	0.18	0.07	0.10	0.04	0.29	0.07	0.00	0.00	0.00	0.13	0.07	0.09	0.67	0.14	0.24
deepseek-r1:8b	0.00	0.00	0.00	0.07	0.29	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.56	0.18	0.27
gemma:7b	0.08	0.18	0.11	0.01	0.18	0.01	0.00	0.07	0.01	0.02	0.07	0.04	0.13	0.07	0.09
mistral:7b	0.14	0.11	0.12	0.03	0.36	0.06	0.00	0.00	0.00	0.06	0.07	0.07	0.09	0.07	0.08
llama3.2:3b	0.11	0.07	0.09	0.01	0.14	0.02	0.00	0.00	0.00	0.12	0.07	0.09	0.00	0.00	0.00
gpt-4o-mini	0.14	0.14	0.14	0.21	0.14	0.17	0.14	0.25	0.18	0.15	0.14	0.14	0.40	0.21	0.28
Average	0.15	0.09	0.10	0.06	0.24	0.08	0.01	0.03	0.02	0.08	0.08	0.07	0.33	0.11	0.16

Model	Threat														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.69	0.39	0.50	0.19	0.78	0.31	0.20	0.17	0.19	0.58	0.30	0.40	0.40	0.26	0.32
deepseek-r1:14b	0.57	0.57	0.57	0.25	0.65	0.36	0.00	0.00	0.00	0.60	0.52	0.56	0.40	0.09	0.14
mistral-nemo:12b	0.50	0.22	0.30	0.20	0.61	0.30	0.06	0.04	0.05	0.63	0.22	0.32	0.75	0.13	0.22
gemma2:9b	0.33	0.43	0.38	0.23	0.65	0.34	0.00	0.00	0.00	0.34	0.43	0.38	0.48	0.48	0.48
llama3.1:8b	0.29	0.26	0.27	0.22	0.52	0.31	0.17	0.09	0.11	0.27	0.13	0.18	0.83	0.22	0.34
deepseek-r1:8b	1.00	0.04	0.08	0.17	0.48	0.25	0.00	0.00	0.00	0.63	0.22	0.32	0.80	0.17	0.29
gemma:7b	0.04	0.22	0.07	0.08	0.52	0.14	0.08	0.48	0.14	0.08	0.48	0.13	0.30	0.57	0.39
mistral:7b	0.00	0.00	0.00	0.04	0.52	0.08	0.06	0.22	0.09	0.00	0.00	0.00	0.50	0.04	0.08
llama3.2:3b	0.25	0.09	0.13	0.07	0.61	0.13	0.15	0.09	0.11	0.20	0.04	0.07	1.00	0.04	0.08
gpt-4o-mini	0.50	0.43	0.47	0.44	0.52	0.48	0.38	0.65	0.48	0.61	0.48	0.54	0.61	0.61	0.61
Average	0.42	0.27	0.28	0.19	0.59	0.27	0.11	0.17	0.12	0.39	0.28	0.29	0.61	0.26	0.29

Model	Insulting														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.00	0.00	0.00	0.29	0.10	0.15	0.09	0.83	0.17	1.00	0.01	0.02	0.54	0.04	0.07
deepseek-r1:14b	0.00	0.00	0.00	0.30	0.08	0.13	0.07	0.64	0.12	0.00	0.00	0.00	1.00	0.01	0.02
mistral-nemo:12b	0.50	0.01	0.01	0.20	0.02	0.03	0.11	0.82	0.19	1.00	0.01	0.01	0.44	0.02	0.04
gemma2:9b	0.33	0.03	0.05	0.19	0.14	0.16	0.09	0.84	0.16	0.39	0.04	0.07	0.29	0.08	0.13
llama3.1:8b	0.00	0.00	0.00	0.25	0.06	0.10	0.05	0.95	0.10	0.00	0.00	0.00	0.00	0.00	0.00
deepseek-r1:8b	0.00	0.00	0.00	0.18	0.10	0.13	0.09	0.68	0.16	0.17	0.01	0.01	0.29	0.01	0.02
gemma:7b	0.33	0.01	0.02	0.20	0.09	0.12	0.04	0.73	0.08	0.00	0.00	0.00	0.33	0.04	0.07
mistral:7b	0.00	0.00	0.00	0.09	0.27	0.14	0.10	0.77	0.17	0.00	0.00	0.00	0.32	0.04	0.07
llama3.2:3b	0.00	0.00	0.00	0.02	0.02	0.02	0.05	0.96	0.09	0.00	0.00	0.00	0.00	0.00	0.00
gpt-4o-mini	0.38	0.19	0.25	0.28	0.30	0.29	0.26	0.28	0.27	0.45	0.19	0.27	0.35	0.16	0.22
Average	0.15	0.02	0.03	0.20	0.12	0.13	0.09	0.75	0.15	0.30	0.03	0.04	0.36	0.04	0.07

Model	Entitlement														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.07	0.01	0.02	0.04	0.14	0.07	0.02	0.04	0.03	0.00	0.00	0.00	0.25	0.01	0.03
deepseek-r1:14b	0.00	0.00	0.00	0.05	0.03	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
mistral-nemo:12b	0.08	0.06	0.07	0.04	0.33	0.08	0.03	0.07	0.04	0.05	0.04	0.05	0.26	0.07	0.11
gemma2:9b	0.18	0.04	0.07	0.05	0.14	0.07	0.05	0.07	0.06	0.13	0.03	0.05	0.23	0.04	0.07
llama3.1:8b	0.04	0.07	0.05	0.04	0.30	0.07	0.03	0.09	0.05	0.04	0.09	0.05	0.17	0.01	0.03
deepseek-r1:8b	0.00	0.00	0.00	0.06	0.19	0.09	0.09	0.01	0.02	0.00	0.00	0.00	0.25	0.01	0.03
gemma:7b	0.03	0.04	0.04	0.01	0.03	0.02	0.00	0.03	0.01	0.03	0.01	0.02	0.24	0.06	0.09
mistral:7b	0.11	0.03	0.05	0.05	0.16	0.07	0.02	0.03	0.03	0.09	0.01	0.02	0.20	0.03	0.05
llama3.2:3b	0.22	0.03	0.05	0.08	0.10	0.09	0.17	0.03	0.05	0.75	0.04	0.08	0.11	0.03	0.04
gpt-4o-mini	0.10	0.04	0.06	0.12	0.17	0.14	0.10	0.19	0.13	0.15	0.04	0.07	0.20	0.07	0.11
Average	0.08	0.03	0.04	0.06	0.16	0.07	0.05	0.06	0.04	0.12	0.03	0.03	0.19	0.03	0.06

Model	Mocking														
	Zero-shot			One-shot			Few-shot			Auto-CoT			Role-based		
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
phi4:14b	0.15	0.50	0.23	0.18	0.29	0.22	0.00	0.00	0.00	0.21	0.28	0.24	0.18	0.47	0.26
deepseek-r1:14b	0.19	0.14	0.16	0.16	0.36	0.22	0.00	0.00	0.00	0.21	0.13	0.16	0.21	0.30	0.25
mistral-nemo:12b	0.22	0.41	0.29	0.21	0.13	0.16	0.00	0.00	0.00	0.20	0.39	0.26	0.20	0.34	0.25
gemma2:9b	0.18	0.41	0.25	0.16	0.37	0.22	0.00	0.00	0.00	0.24	0.27	0.25	0.25	0.39	0.30
llama3.1:8b	0.17	0.35	0.23	0.12	0.30	0.17	0.00	0.00	0.00	0.31	0.14	0.19	0.12	0.50	0.19
deepseek-r1:8b	0.00	0.00	0.00	0.15	0.19	0.17	0.00	0.00	0.00	0.19	0.30	0.23	0.13	0.35	0.19
gemma:7b	0.10	0.44	0.16	0.11	0.36	0.16	0.00	0.00	0.00	0.07	0.60	0.13	0.17	0.24	0.20
mistral:7b	0.26	0.08	0.12	0.15	0.21	0.18	0.00	0.00	0.00	0.29	0.08	0.12	0.24	0.15	0.19
llama3.2:3b	0.08	0.69	0.14	0.10	0.15	0.12	0.00	0.00	0.00	0.15	0.26	0.19	0.07	0.81	0.13
gpt-4o-mini	0.24	0.27	0.25	0.25	0.30	0.27	0.29	0.33	0.31	0.22	0.28	0.25	0.26	0.28	0.27
Average	0.16	0.33	0.18	0.16	0.27	0.19	0.03	0.03	0.03	0.21	0.27	0.20	0.18	0.38	0.22

other strategies, yielding significantly higher scores in both F1 and Precision across multiple incivility categories.

> **Finding 4:** *Role-based* prompting outperformed other strategies in fine-grained incivility detection, achieving higher precision and F1-scores. In contrast, the *few-shot* strategy ($k = 3$) showed the worst results. Furthermore, the gaps between precision and recall suggest the conservative behavior of SLM models.

4.3 SLMs vs. ML+NLP Models Comparison (RQ$_3$)

To answer **RQ$_3$**, we compare the performance of the best SLM configuration against five ML models – Multinomial Naive Bayes (MNB), Logistic Regression Classifier (LRC), Random Forest Classifier (RFC), AdaBoost Classifier (ABC), and DistilBERT (DBERT) – combined with two NLP techniques: Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). We select the best SLM configuration based on F1-score (F1) and Precision (Pr) for both incivility levels. To this end, we analyze which configurations tend to appear more times in the top-3 by F1 and Pr. At the coarse-grained level, *gpt-4o-mini* appears two times, and at the fine-grained level, *deepseek-r1:14b* appears 5 times, both with the *role-based* strategy, achieved the best overall performance.

The best SLM configuration vs. ML models+NLP techniques by incivility level. Tables 11 and 12 show the performance score differences between the best SLM configurations – *gpt-4o-mini* (coarse-grained) and *deepseek-r1:14b* (fine-grained) – and five ML models. Cells highlighted in Green indicate a positive difference ($\Delta > 0$), while those in Red indicate a negative difference ($\Delta < 0$).

Table 11 shows that for uncivil comments, the *gpt-4o-mini* outperforms most ML models combined with NLP techniques. Only the *DistilBERT* model achieves higher scores in terms of Recall (Re) and F1, while still underperforming in Pr. For the civil comments, *gpt-4o-mini* outperforms all ML models in terms of F1. However,

Table 11: Delta of best SLM (gpt-4o-mini + role-based, coarse-grained) vs. ML+NLP models

Best SLM Configuration (`gpt-4o-mini` + role based prompting)						
Model	Civil			Uncivil		
	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1
ABC + BoW	0.09	-0.01	0.06	0.17	0.36	0.32
LRC + BoW	0.09	0.01	0.06	0.21	0.34	0.31
MNB + BoW	0.08	0.12	0.11	0.31	0.22	0.26
RFC + BoW	0.09	-0.01	0.05	0.15	0.37	0.33
ABC + TF-IDF	0.07	0.36	0.24	0.38	0.00	0.21
LRC + TF-IDF	0.13	-0.08	0.05	0.25	0.59	0.64
MNB + TF-IDF	0.12	-0.05	0.05	0.12	0.50	0.50
RFC + TF-IDF	0.12	-0.06	0.05	0.15	0.52	0.53
DBERT	-0.01	0.02	0.01	0.04	-0.03	-0.01

its Re is lower than *ABC + BoW*, *RFC + BoW*, *LRC + TF-IDF*, *MNB + TF-IDF*, and *RFC + TF-IDF*, while still outperforming the remaining ML configurations. In terms of Pr, only *DBERT* slightly outperforms *gpt-4o-mini*, with a marginal difference of 0.01. These results suggest that *gpt-4o-mini*, when combined with a role-based prompting strategy, provides a competitive performance, particularly by maintaining a strong balance between Precision and F1 in both classes. This highlights its potential as an effective alternative to traditional ML pipelines for coarse-grained classification tasks.

> **Finding 5:** The *gpt-4o-mini*, when guided by role-based prompting, outperforms ML models combined with NLP techniques in the detection of civil and uncivil comments.

Table 12 shows that, despite the overall lower performance, the *deepseek-r1:14b* still outperforms ML models across most incivility types. In particular, the combination of *LRC + BoW* outperforms SLM in terms of F1 ($\Delta < 0$) for the *Threat*, *Insulting*, and *Entitlement* incivilities, indicating that ML models can still be competitive for incivility types that are more explicit or clearly defined. A similar pattern is observed in the *None* category, in which all ML models outperform *deepseek-r1:14b* in terms of Recall. This suggests that, for incivilities that are more explicit, simpler models may still be effective when combined with appropriate feature representations.

Table 12: Delta of best SLM (deepseek-r1:14b + role-based, fine-grained) vs. ML+NLP models

Model	Bitter Frustration			Impatience			Vulgarity			Irony			Identify Attack		
	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1
ABC+ BoW	0.24	0.61	0.34	0.41	0.06	0.11	0.67	0.11	0.19	0.00	0.00	0.00	0.33	0.04	0.06
LRC + BoW	0.08	0.48	0.20	0.33	0.02	0.06	0.53	0.05	0.11	0.00	0.00	0.00	0.33	0.04	0.06
MNB + BoW	0.14	0.53	0.25	0.41	0.06	0.11	0.67	0.11	0.19	0.00	0.00	0.00	0.33	0.04	0.06
RFC + BoW	0.24	0.61	0.34	0.41	0.06	0.11	0.57	0.10	0.17	0.00	0.00	0.00	0.33	0.04	0.06
ABC + TF-IDF	0.24	0.61	0.34	0.41	0.06	0.11	0.67	0.11	0.19	0.00	0.00	0.00	0.33	0.04	0.06
LRC + TF-IDF	-0.01	0.56	0.26	0.26	0.05	0.10	0.57	0.10	0.17	0.00	0.00	0.00	0.33	0.04	0.06
MNB + TF-IDF	0.24	0.61	0.34	0.41	0.06	0.11	0.67	0.11	0.19	0.00	0.00	0.00	0.33	0.04	0.06
RFC + TF-IDF	0.24	0.61	0.34	0.41	0.06	0.11	0.67	0.11	0.19	0.00	0.00	0.00	0.33	0.04	0.06
DBERT	-0.04	0.28	0.03	0.18	-0.01	0.00	0.18	-0.24	-0.21	0.00	0.00	0.00	0.33	0.04	0.06
Model	**Threat**			**Insulting**			**Entitlement**			**Mocking**			**None**		
	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1	Δ-Pr	Δ-Re	Δ-F1
ABC+ BoW	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.21	0.30	0.25	0.13	-0.15	0.00
LRC + BoW	0.15	-0.04	-0.03	0.86	-0.08	-0.09	-0.08	-0.04	-0.05	0.11	0.25	0.19	0.09	-0.06	0.02
MNB + BoW	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	-0.02	0.28	0.21	0.12	-0.09	0.02
RFC + BoW	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.17	0.28	0.22	0.13	-0.14	0.00
ABC + TF-IDF	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.21	0.30	0.25	0.13	-0.15	0.00
LRC + TF-IDF	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.18	0.29	0.24	0.12	-0.14	0.00
MNB + TF-IDF	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.21	0.30	0.25	0.13	-0.15	0.00
RFC + TF-IDF	0.40	0.09	0.14	1.00	0.01	0.02	0.00	0.00	0.00	0.17	0.28	0.23	0.13	-0.14	0.00
DBERT	0.40	0.09	0.14	0.73	-0.11	-0.14	0.00	0.00	0.00	-0.02	0.18	0.10	0.05	-0.09	-0.03

Additionally, for the *Irony* incivility, both the SLM and ML models achieve a score of 0.00, resulting in a performance difference of (Δ = 0). We emphasize that several ML models obtained zero scores for precision, recall, and F1 in multiple types of incivility, which explains the high Δ on Table 12. For instance, for *Bitter Frustration*, both AdaBoost and Random Forest under both TF-IDF and Bag-of-Words representations failed to identify any instances correctly.

Finding 6: *Deepseek-r1:14b* often outperforms traditional ML models. ML models like *LRC + BoW* still achieve higher performance in more explicit categories such as *Threat*, *Insulting*, and *Entitlement*. This implies that ML models, combined with effective feature representations, remain competitive for certain types of incivility.

5 Study Implications

We present four implications for both researchers and tool builders.

Understanding the effectiveness of SLMs in detecting incivility at different levels of granularity. Our results show that SLMs can effectively classify incivility at a coarse-grained level (i.e., distinguishing between *civil* and *uncivil* comments). Specifically, SLMs with 9B+ parameters demonstrate robust performance on uncivil comments, making them strong candidates for real-world application to moderate incivility in GitHub conversations. Nevertheless, at a fine-grained level (i.e., different types of incivility), the SLMs exhibit considerable variation in performance across types of incivility, especially certain types of incivility that are implicit in nature. e.g., *Irony*, *Impatience*, *Mocking*, and *Vulgarity*. These findings suggest important takeaways for researchers and tool builders: (i) there is a need for more focused approaches, such as fine-tuning SLM models on domain-specific data or integrating contextual and conversational cues to detect or avoid incivility; (ii) these strategies can enhance the SLM model's ability to detect specific types of incivility, particularly in collaborative software development environments, like code reviews performed on GitHub.

Understanding the influence of different prompt strategies on incivility detection. Our findings emphasize the pivotal role of prompt design in enhancing the effectiveness of SLMs. Specifically, the analysis in RQ2 and RQ3 reveals that *role-based prompting* consistently achieves the highest performance among all evaluated prompt strategies. This indicates that assigning a defined role or perspective to the model significantly enhances its ability to accurately classify incivility at different levels. These insights suggest two additional takeaways: (i) the use of carefully crafted prompts that simulate key roles (e.g., developer, moderator, and reviewer). helps SLM models to improve the detection of incivility comments, especially those that are implicit in nature; and (ii) integrating role-based prompts into incivility detection tools can enhance their effectiveness, ensuring that the tools are more sensitive to context-specific and social dynamics in collaborative environments.

Enabling automatic support for moderating incivility in GitHub conversations. The need for effective tools to moderate incivility in GitHub conversations has become critical as GitHub grows as a collaborative development platform. The current solutions offer some level of incivility detection, but tend to fall in terms of precision, especially in the case of implicit types of incivility, such as *Irony*, *Impatience*, or *Mocking*. In this case, the development of automated tools specifically tailored for GitHub conversations would allow for a more accurate and efficient identification of inappropriate or uncivil behavior, creating a healthier and more productive collaborative environment. For instance, such tools could be integrated into pull request or issue threads to automatically flag comments with potentially uncivil tone, provide real-time feedback to reduce or avoid incivility. These features would help the maintainers and contributors to address incivility early and maintain constructive collaboration dynamics.

Understanding the trade-off between precision and recall is essential when defining the system's goals. Our findings offer empirical support for making informed decisions regarding the trade-off between precision and recall. A language model optimized for high precision tends to be more conservative; it minimizes false positives but may potentially fail to detect implicit types of incivility, such as *Irony*. In contrast, prioritizing recall enhances the model's ability to capture a wider range of uncivil behaviors, including implicit instances, and helps reduce the risk of missing severe cases like *Identity Attacks*. However, this approach may increase false positives by misclassifying neutral comments as offensive. Striking the right balance between precision and recall requires careful consideration of the deployment context and associated risks.

6 Threats to Validity

We discuss threats to the study validity [33] as follows.

Construct Validity. We used well-known evaluation metrics such as precision, recall, and F1-score to assess model performance. However, the use of manual prompt engineering may introduce a degree of subjectivity, as slight variations in phrasing can affect outcomes. We mitigated this by iteratively designing and refining the prompts based on initial model responses to ensure clarity, relevance, and alignment with our study goal. Moreover, we make all prompts publicly available to promote transparency and reproducibility. We also avoided fine-tuning SLMs and ML models on incivility data. This decision reflects a growing interest in exploring the out-of-the-box capabilities of SLMs, prioritizing their applicability and computational efficiency. Additionally, our findings demonstrate that good results can be achieved without fine-tuning.

Internal Validity. We relied on two datasets [7, 24], chosen for their robustness and relevance in prior research. We acknowledge that limitations on data quality and labeling could have introduced biases that affected the evaluation results. However, the use of these datasets provides a common ground for comparison of our findings with future works. Regarding potential data contamination, the deduplication against pre-training corpora is infeasible due to model and data opacity. While we cannot rule out overlap with our GitHub comments, this risk is mitigated by the nature of our task (classification, not generation) and the absence of verbatim outputs in manual inspection. Moreover, incivility instances were processed sequentially within the same session, which may have allowed unintended context transfer between datasets. Additionally, we explored only a limited set of prompting strategies. Given the rapid development of prompting techniques, future work should consider various strategies to enhance evaluation robustness.

Conclusion Validity. The differences in model architectures, training data, and reasoning capabilities may have contributed to the observed performances. To strengthen our conclusions, we ensured a fair comparison by applying the same prompts and evaluation criteria across all models, including both SLMs and ML models. This consistency helps reduce threats to inference validity and supports the credibility of our findings. Additionally, we apply the Friedman non-parametric test with Nemenyi post-hoc analysis to avoid subjectivity in comparing SLMs and prompt strategies.

External Validity. Our findings are specific to incivility detection in GitHub conversations, and may not generalize to other online collaborative platforms (e.g., Stack Overflow, and Reddit). Moreover, the models were evaluated in English, and their effectiveness may differ in multilingual or culturally diverse environments where expressions of incivility vary. Finally, we relied on pre-existing datasets with predefined categories of incivility, which may not capture the full spectrum of toxic or harmful behaviors in real-world scenarios. Despite these limitations, we compared SLMs with traditional ML models and observed that, even without fine-tuning, SLMs guided by role-based prompts can achieve competitive performance. This suggests promising potential for practical deployment, especially in resource-constrained settings. Future research across different domains, languages, and platforms is needed to further establish the external validity of our approach.

7 Related Work

The concept of *incivility*, though rooted in communication studies [5], has been gradually adopted by the software engineering community to address disruptive behaviors in collaborative development environments. In software engineering, incivility is increasingly examined through the lens of peer review and issue tracking platforms. For instance, Rahman et al. [24] investigated incivility in code review discussions by developing a classification model and curating a dataset of 1,000 manually annotated comments. This dataset is particularly useful for coarse-grained classification tasks (and was directly utilized by our study). Similarly, Ferreira et al. [10] leveraged Coe et al. [5]'s incivility framework to propose the Tone Bearing Discussion Features (Tbdf) for classifying uncivil behaviors in code reviews, which served as the foundation for our fine-grained incivility classes. Building on this, Ferreira et al. [9] further explored the dynamics of heated conversations in GitHub locked issues, providing valuable insights into the manifestation of incivility in community governance contexts.

In addition to research explicitly focused on incivility, several studies have addressed similar interpersonal issues using alternative terminologies, particularly *toxicity*. These works examine problematic communication behaviors through lenses like aggression, disrespect, and hostility, often aligning conceptually with incivility. A notable example is the work by Miller et al. [19], which studied toxic behaviors in open-source discussions. Their analysis highlighted how missed responses and confrontational tones can deteriorate collaborative environments. Despite the different terminology, such studies enrich the discourse on developer interaction quality. Thus, even though the term *toxicity* carries slightly different theoretical assumptions, its empirical overlap with incivility makes it a valuable construct in comparative analyses.

Our study design directly builds upon the conceptual and empirical groundwork laid by these previous studies. Specifically, we adopted the 10 incivility classes proposed by Ehsani et al. [7], which consolidated data from three other works: Ferreira et al. [10], Sarker et al. [28], and Miller et al. [19]. These classes are listed in Table 2. They were used for our classification processes at the fine-grained level. Aside from the classes, we also utilized the dataset by Ehsani et al. [7], which includes 5,961 labeled comments from GitHub discussions, as the ground truth for our fine-grained experiments. Furthermore, we utilized a merged dataset, that utilizes a subset of these instances, combined with Rahman et al. [24] civil/uncivil dataset, to support our coarse-grained classification tasks. These foundational resources facilitated robust comparative evaluations.

8 Conclusion and Future Work

In this paper, we investigated the effectiveness of 10 Small Language Models (SLMs) in detecting incivility in GitHub conversations at two levels of granularity: coarse-grained (civil vs. uncivil) and fine-grained (specific types of incivility). Our findings underscore the importance of prompt engineering in harnessing SLM capabilities. Among the strategies, *role-based prompting* consistently delivered the best results for both coarse- and fine-grained incivility. SLMs with over 9B parameters generally outperformed smaller models in detecting uncivil comments. However, both SLMs and traditional ML models struggled with implicit incivilities, highlighting the challenge of identifying nuanced language in GitHub discussions. Overall, SLMs, depending on size and prompting, can surpass ML models. Our findings also support the development of non-intrusive, context-aware moderation tools powered by LLMs or SLMs for platforms like GitHub. As future work, we plan to explore *hybrid prompting strategies*, such as combining auto-CoT with *role-based prompting*, to further improve SLM performance. We also plan to investigate how the contextual information within conversations affects the effectiveness of SLMs in detecting incivility.

ARTIFACT AVAILABILITY

All artifacts, including datasets, scripts, and documentation from this study are available at https://doi.org/10.5281/zenodo.16998718.

ACKNOWLEDGMENTS

This work was partially funded by CAPES (88881.879016/2023-01), CNPq (404406/2023-8, 310391/2025-3, 315711/2020-5), FAPERJ (200.510/2023, 211.033/2019, 202.621/2019), FAPESP (2023/00811-0), and FUNCAP (BP6-00241-00276.01.00/25).

REFERENCES

[1] Caio Barbosa, Anderson Uchôa, Daniel Coutinho, Wesley KG Assunção, Anderson Oliveira, Alessandro Garcia, Baldoino Fonseca, Matheus Rabelo, José Eric Coelho, Eryka Carvalho, et al. 2023. Beyond the Code: Investigating the Effects of Pull Request Conversations on Design Decay. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 1–12.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[3] Victor R Basili-Gianluigi Caldiera and H Dieter Rombach. 1994. Goal question metric paradigm. *Encyclopedia of software engineering* 1, 528-532 (1994), 6. doi:~mvz/handouts/gqm.pdf

[4] Benjamin Clavié, Alexandru Ciceu, Frederick Naylor, Guillaume Soulié, and Thomas Brightwell. 2023. Large language models in the workplace: A case study on prompt engineering for job type classification. In *International conference on applications of natural language to information systems*. Springer, 3–17.

[5] Kevin Coe, Kate Kenski, and Stephen A Rains. 2014. Online and uncivil? Patterns and determinants of incivility in newspaper website comments. *Journal of communication* 64, 4 (2014), 658–679.

[6] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, Vol. 11. 512–515.

[7] Ramtin Ehsani, Mia Mohammad Imran, Robert Zita, Kostadin Damevski, and Preetha Chatterjee. 2024. Incivility in open source projects: A comprehensive annotated dataset of locked github issue threads. In *Proceedings of the 21st International Conference on Mining Software Repositories*. 515–519.

[8] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[9] Isabella Ferreira, Bram Adams, and Jinghui Cheng. 2022. How heated is it? Understanding GitHub locked issues. In *Proceedings of the 19th International Conference on Mining Software Repositories*. 309–320.

[10] Isabella Ferreira, Jinghui Cheng, and Bram Adams. 2021. The" shut the f** k up" phenomenon: Characterizing incivility in open source code review discussions. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–35.

[11] Isabella Ferreira, Ahlaam Rafiq, and Jinghui Cheng. 2024. Incivility detection in open source code review and issue discussions. *Journal of Systems and Software* 209 (2024), 111935.

[12] Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32, 200 (1937), 675–701.

[13] Daviti Gachechiladze, Filippo Lanubile, Nicole Novielli, and Alexander Serebrenik. 2017. Anger and its direction in collaborative software development. In *2017 IEEE/ACM 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER)*. IEEE, 11–14.

[14] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology* 33, 8 (2024), 1–79.

[15] Ron Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Vol. 14. Montreal, Canada, 1137–1145.

[16] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2023. Better zero-shot reasoning with role-play prompting. *arXiv preprint arXiv:2308.07702* (2023).

[17] Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better Zero-Shot Reasoning with Role-Play Prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 4099–4113. doi:10.18653/v1/2024.naacl-long.228

[18] Mika Mäntylä, Bram Adams, Giuseppe Destefanis, Daniel Graziotin, and Marco Ortu. 2016. Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?. In *Proceedings of the 13th international conference on mining software repositories*. 247–258.

[19] Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian KaUstner. 2022. " Did you miss my comment or what?" understanding toxicity in open source discussions. In *Proceedings of the 44th International Conference on Software Engineering*. 710–722.

[20] OpenAI. 2024. Gpt-4o mini: advancing costefficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. (Accessed on 14/03/2025).

[21] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. 2017. Deep Learning for User Comment Moderation. In *Proceedings of the First Workshop on Abusive Language Online, ALW@ACL 2017, Vancouver, BC, Canada, August 4, 2017*, Zeerak Waseem, Wendy Hui Kyong Chung, Dirk Hovy, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 25–35. doi:10.18653/V1/W17-3004

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[23] Mohammad Masudur Rahman and Chanchal K Roy. 2014. An insight into the pull requests of github. In *Proceedings of the 11th working conference on mining software repositories*. 364–367.

[24] Md Shamimur Rahman, Zadia Codabux, and Chanchal K Roy. 2024. Do words have power? understanding and fostering civility in code review discussion. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 1632–1655.

[25] Matthew Renze. 2024. The Effect of Sampling Temperature on Problem Solving in Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 7346–7356. doi:10.18653/v1/2024.findings-emnlp.432

[26] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927* (2024).

[27] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[28] Jaydeb Sarker, Asif Kamal Turzo, Ming Dong, and Amiangshu Bosu. 2023. Automated identification of toxic code reviews using toxicr. *ACM Transactions on Software Engineering and Methodology* 32, 5 (2023), 1–32.

[29] Timo Schick and Hinrich Schütze. 2021. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 2339–2352. doi:10.18653/V1/2021.NAACL-MAIN.185

[30] Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21.

[31] Igor Steinmacher, Tayana Conte, Marco Aurélio Gerosa, and David Redmiles. 2015. Social barriers faced by newcomers placing their first contribution in open source software projects. In *Proceedings of the 18th ACM conference on Computer supported cooperative work & social computing*. 1379–1392.

[32] Parastou Tourani, Bram Adams, and Alexander Serebrenik. 2017. Code of conduct in open source projects. In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)*. IEEE, 24–33.

[33] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.

[34] Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*. Springer, 745–760.

[35] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493* (2022).

[36] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. https://openreview.net/forum?id=WZH7099tgfM

[37] Haiyi Zhu, Robert Kraut, and Aniket Kittur. 2012. Effectiveness of shared leadership in online communities. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. 407–416.