

# Prompt-driven Detection of Offensive Urdu Language using Large Language Models

Iffat Maab<sup>1</sup>, Usman Haider<sup>2</sup>, Junichi Yamagishi<sup>1</sup>

<sup>1</sup>National Institute of Informatics, Tokyo, Japan

<sup>2</sup>University of Galway, Ireland

{maab, jyamagis}@nii.ac.jp

usman.haider@universityofgalway.ie

## Abstract

Offensive language detection poses a significant challenge in modern social spaces, necessitating advanced solutions. Online media platforms have been known to escalate acts of violence and broader conflicts, and thus, an automated system to help counter offensive content is essential. Traditional NLP models have typically dominated the field of hate speech detection, but require careful model design and extensive tuning. Moreover, a notable resource gap exists for addressing offensive languages, particularly those transcribed in non-native scripts, such as Roman Urdu and Urdu. This study explores the potential of pre-trained LLMs in using prompt-based methods using different transcriptions of the Urdu language, particularly their efficacy in detecting offensive content in diverse linguistic contexts. Our study employs state-of-the-art open-source LLMs, including advanced variants of Llama, Qwen, Lughaat, and proprietary GPT-4, which are evaluated through prompting strategies in different under-resourced languages. Our findings show that pre-trained LLMs achieve performance comparable to traditional fine-tuned benchmarks in detecting hateful and offensive content.

**Content Warning:** This paper contains examples from datasets which are offensive in nature.

## 1 Introduction

In today's digital age, social computing has significantly reshaped how people interact, particularly on social media platforms and online forums. However, these same modalities create a conducive environment for the spread of harmful and aggressive expressions. Detecting hate speech is challenging for humans (Jackson, 2011), which highlights the critical need to fully understand the strengths and limitations of crowd-sourced annotation, not only in English but also in under-resourced languages.

Developing and evaluating hate speech detection systems for low-resource languages is a useful and understudied direction, which requires the creation of annotated datasets (Ullah et al., 2023).

Recent advancements in large language models (LLMs) have demonstrated significant improvements in a variety of NLP tasks through zero-shot and few-shot learning techniques (Brown et al., 2020; Kojima et al., 2022). Notably, in-context learning, which does not require fine-tuning, has significantly advanced the capabilities of these models (Meta, 2024; Achiam et al., 2023; Touvron et al., 2023; Beltagy et al., 2022; Shin et al., 2020; Schick and Schütze, 2020). However, these developments primarily benefit resource-rich languages, while under-resourced languages such as Urdu still face considerable challenges in terms of accuracy and reliability. Urdu, spoken by more than 70 million people, has a complex morphological structure and remains significantly underrepresented in NLP (Blasi et al., 2021). This presents distinct challenges in evaluating LLMs performance in generation tasks for this language (Arif et al., 2024).

Although English serves as the official language of Pakistan (Rahman, 1997), Urdu holds the status of the national language and plays a central role in cultural and social interactions. The Urdu script is an abjad writing system that evolved from the Persian script, which in turn originated from traditions of the Arabic script (Alam and Hussain, 2022; Ubul et al., 2017). In everyday written communication, it is common for individuals to use Romanized Urdu, where the Urdu language is transcribed using Latin characters (Shakeel and Karim, 2020). People frequently engage in code-switching, seamlessly alternating between English and Urdu at various linguistic levels, such as within a single sentence, clause, or even at the phrase level, reflecting the bilingual and dynamic nature of communication in Pakistan (Ansari et al., 2020; Bashir, 2011;

Garcia, 2011). This poses challenges as Roman Urdu is an informal and colloquial language that does not adhere to standardized word forms, sentence structure, uniform abbreviations, or specific spelling rules (Saeed et al., 2021), thus presenting a substantially greater modeling challenge.

Supervised deep learning methods currently represent the cutting-edge approach for text classification tasks, including online hate speech detection (Nobata et al., 2016; Van Aken et al., 2018; Akhter et al., 2020; Rizwan et al., 2020). Although fine-tuned models are renowned for their strong performance in a variety of tasks, the complexity and resource requirements of fine-tuning these models tend to make them less adaptable and more prone to dataset biases (Plaza-del Arco et al., 2021; Parthasarathy et al., 2024; Huang et al., 2020). Similarly, the reliance on manually annotated datasets in hate speech detection systems has led studies (Waseem and Hovy, 2016; Kwok and Wang, 2013) to demonstrate that a narrow focus on specific types of hate speech or scenarios can skew their content (Gao et al., 2017).

To the best of our knowledge, the literature on in-context learning through prompt engineering for Urdu text classification is quite sparse, with only a few studies addressing this topic (Ullah et al., 2023; Arif et al., 2024; Rizwan et al., 2020). Taken together, Rizwan et al. (2020) contributes the RUHSOLD dataset and establishes supervised baselines using pretrained embeddings, providing a valuable resource for hate speech detection in Roman Urdu. Ullah et al. (2023) compared standard fine-tuning and prompt-based fine-tuning with BERT-family models, offering insights into parameter-efficient adaptation for Urdu text classification. More recently, Arif et al. (2024) explores the applicability of generalist models such as GPT-4 and Llama-8B, using Urdu-script dataset. In contrast, our work centers on inference-time prompting and in-context learning with modern LLMs, extending existing evaluations across scripts and datasets to better understand their behavior in low-resource and multilingual settings. This paper investigates the effectiveness of prompts in English, Roman Urdu, and Urdu to identify offensive language within comments sourced from both Roman Urdu (Rizwan et al., 2020; Akhter et al., 2020) and Urdu (Arif et al., 2024) texts on social platforms such as Twitter and YouTube. In this manuscript, we focus on the detection of offensive Urdu language and use

PER-CLASS Labels	RUHSOLD		RUTUBE	USADC
	Coarse	Fine	Binary	Binary
Normal	5342	4917	5000	2816
Abusive	4650	2208	5000	2854
Sexual Hate	-	770	-	-
Religious Hate	-	719	-	-
Profane	-	582	-	-
<b>Prompts Tested</b>				
English	✓	✓	✓	✓
Roman Urdu	✓	✓	-	-
Urdu	✓	-	-	-

Table 1: Summary of the data available for coarse- and fine-grained tasks using RUHSOLD, and binary tasks for RUTUBE and USADC, while also detailing the implementation of prompt-based methodologies across each dataset.

the term abusive interchangeably with offensive, acknowledging their nuanced differences. This choice aligns with previous studies and datasets (Rizwan et al., 2020; Akhter et al., 2020), which often merge these categories under terms like "offensive/abusive," so we adopt this convention for consistency with the existing literature and resources.

In our study, we first aim to determine how well prompt-based approaches with small and large pretrained LLMs capture the complex linguistic nuances of Urdu in detecting hate speech and abusive texts to clarify their potential for under-resourced languages with limited labeled datasets. Then, we assess the performance improvements offered by different prompt-based Urdu transcriptions as shown in Table 7 in the Appendix to assess which models most effectively understand Urdu in its various forms and scripts. To this end, we conduct an extensive evaluation of multiple open-source LLMs on coarse- and fine-grained classification tasks for Roman Urdu and Urdu datasets. This approach not only benchmarks the performance of different linguistic prompts, but also enhances our understanding of language processing in bilingual contexts.

## 2 Related work

Research on automated detection of hate speech has advanced significantly in the past five years. This area comprises a variety of interconnected tasks. Waseem and Hovy (2016) provides a dataset comprising 16,000 hate speech-labeled tweets, exploring the most effective features for accurate identification. In a subsequent study, Waseem (2016)

expanded on this work by addressing personal bias in annotations through the involvement of multiple annotators. An important contribution to the field was the development of Roman Urdu hate speech and offensive language dataset called RUHSOLD by Rizwan et al. (2020). To advance the detection of hate speech and offensive language in multiple languages, the Forum for Information Retrieval Evaluation (FIRE) hosts an initiative known as Hate-Speech and Offensive Content Identification (HASOC) in Indo-European languages (Mandl et al., 2019). Akhter et al. (2020) also evaluates different machine learning classifiers to detect offensive language within Urdu text.

In recent studies, such as the work by An (2023), highlighted the utility of prompt-based learning in low-resource linguistic settings where traditional fine-tuning is computationally expensive. Other studies (Girón et al., 2025; Gao et al., 2021; Wang et al., 2022; Lee et al., 2021; Jin et al., 2022; Schucher et al., 2022) also affirm the superior classification performance of prompt-based methods in environments with limited resources. Designing language-specific prompts for language models significantly improves their ability to make accurate predictions by effectively utilizing contextual cues, even with minimal examples (few-shots) or in zero-shot settings (Seo et al., 2022). Along these lines, Tahir et al. (2024) evaluate the capabilities of LLMs including GPT-3.5-Turbo in a zero-shot setting across 14 different tasks using 15 distinct Urdu datasets. Singh et al. (2024) evaluates state-of-the-art open-source LLMs on generation tasks across 29 under-represented Indic languages. Recent work by Jin et al. (2024) explores the use of GPT-3.5 in generating realistic test cases for hate speech detection to uncover the shortcomings of LLMs. In essence, LLMs offer a wide range of useful applications that exceed their initial generative capabilities.

It is beneficial for the community that users can interpret multiple orthographies that support seamless text conversion (Person, 2009). Bāsāá’s orthographic history reflects diverse systems shaped by English, German, and French transcription practices (Bird, 2001). Recent work by O’Neil et al. (2023) highlights the use of NLP techniques over traditional baselines in resource-scarce environments, focusing on the orthographic conversion of Bāsāá texts from the Protestant missionary orthography to the current national standard, AGLC (Gen-

eral Alphabet for Cameroon Languages). Such efforts demonstrate how standardized Latin-script representations can enhance computations and linguistic accessibility in under-resourced languages, mirroring the role of Roman Urdu as a bridge toward broader NLP support for Urdu. Jaavid et al. (2024) extends LLMs to non-English languages with non-Roman scripts by using romanized text as an interface for instruction tuning. Hermjakob et al. (2018) introduced uroman, a tool designed to convert text from numerous languages and scripts, including Chinese, Arabic, and Cyrillic, into a unified Latin-script format. Adouane et al. (2016) addressed language identification for informal Romanized Arabic and Berber using supervised machine learning to handle mixed-language text.

### 3 Methodology

Previous studies indicate that adapting LLMs through instruction-based fine-tuning is a common method to achieve gains in performance (Davidson et al., 2017; Fortuna and Nunes, 2018). However, fine-tuning language models is resource intensive and is dependent on annotated data. Consequently, efficacy of fine-tuned methods depends significantly on the quality and volume of training data, and creating annotated datasets for toxic content such as racism, attacks, hate speech, abuse, and offense is challenging for various languages.

The methodology adopted in this study is designed to explore how pretrained LLMs in hate speech detection for under-resourced languages perform. Prompt engineering research has mainly focused on English, but recent studies, such as Song et al. (2022), have tested its effectiveness in other languages such as Chinese, with results indicating that prompt usage led to favorable results for text classification. Winata et al. (2021) and Nie et al. (2022) worked on non-English prompts on several multilingual tasks. Our work continues in this direction; we combine in-context learning through different language prompts to determine the capacity of various LLMs in processing the linguistic subtleties of Urdu. The primary objective is to identify the performance gaps of fine-tuned models versus pretrained LLMs, reducing the reliance on model fine-tuning and scarce annotated data for low-resource languages, such as Urdu (Rizwan et al., 2020; Ullah et al., 2023). Our approach increases flexibility in model selection, particularly through the use of multiple LLMs.

The design of our prompts is primarily driven by their inherent flexibility to provide a robust platform for experimenting with various LLMs across different sizes, using different languages, and without the need for fine-tuning. Similar type of prompting methods exist for hate speech classification as reported by [Choudhary et al. \(2025\)](#); [Han and Tang \(2022\)](#) which also shows that a substantial number of input-labels pairs (few-shots) are necessary for good performance. To demonstrate the adaptability of LLMs to the Urdu language across varying linguistic contexts, we introduce prompts in three different settings: first in English, then in Roman Urdu, and finally in Urdu. We evaluate the prompts in the following settings.

**Zero-shot** For zero-shot learning, the model is provided with a natural language query without question-answer pairs for sentence classification and is not fine-tuned. The zero-shot settings are listed at the top in [Table 7](#) in the Appendix.

**Few-shot** (K-SHOTS) In the few-shot setting, relevant examples are provided in addition to the zero-shot prompt, except that the examples are provided in question-answer pairs (i.e., sentence-label pairs). Given the limited data, the parameter ‘k’ (number of examples) is selected based on the specific task. We carefully selected K-SHOTS examples manually, showcasing the best cases for each category. Our K-SHOTS examples were manually curated to ensure balanced and informative prompts. This was necessary because the fine-grained RUHSOLD dataset is highly imbalanced, sentence lengths vary substantially, and random sampling can produce ambiguous or unrepresentative examples. Given that in-context learning for Urdu hate speech is still underexplored, manual selection helped maintain clarity and consistency. We therefore used the same curated k-shot set across English, Roman Urdu, and Urdu prompts, while ensuring coverage of frequently used terms that distinguish abuse categories in each language.

**Prompt Design** Given the toxic nature of datasets, we tested multiple prompts and found the system prompt: "You are an abusive language detector" more effective than "offensive language detector." Also, since fine-grained classification requires distinguishing categories such as sexual hate, religious hate, and profane language, framing the task as abusive language detection proved more appropriate. For the Roman Urdu prompt, we translated

the English version into Latin script while preserving semantic fidelity. Direct adoption, however, required iterative refinements; a particular challenge emerged with the translation of the label "Abusive" for task prediction. The term "Gaali" directly conveys the meaning of abuse, but its antonym "Not abusive" lacks a clear, single context-independent translation in Roman Urdu or Urdu. Its meaning varies by context, with formal or literary alternatives including Tareef karna (praise), Tahsin (appreciation), Shaista kalami (polite speech), and Narm guftugu (gentle conversation). For in-context learning, classification requires labels with direct relevance so that LLMs can consistently map inputs to outputs. After careful analysis of some test prompts, we determined that ‘Ikhlaqi’ (ethical) and ‘Khair Ikhlaqi’ (non-ethical) best operationalize the categories, being both semantically appropriate and prevalent in our test sets. For more prompt details, see [Appendix B](#).

## 4 Experimental Setup

**Datasets** Our study utilizes three distinct datasets that span a range of content related to hate speech, including the detection of abusive language. **The first** dataset is Roman Urdu Hate-Speech and Offensive Language Detection (RUHSOLD) derived from [Rizwan et al. \(2020\)](#), and comprises 9,992 and 9,196 tweets labeled for coarse- and fine-grained classification, respectively. The coarse-grained dataset is categorized into binary labels for Abusive/Offensive versus Normal/Non-Abusive, while the fine-grained dataset expands the Abusive/Offensive label into four granular level categories of Abusive, Religious Hate, Sexual Hate, and Profane/Untargeted. [Table 1](#) provides a summary of the examples available for each task. **The second** dataset is also in Roman Urdu, sourced from [Akhter et al. \(2020\)](#)<sup>1</sup>. From the original 147,000 user comments collected from various YouTube videos, we randomly sampled 10,000 comments to enable a balanced comparison with the RUHSOLD dataset. We refer to this dataset as Roman Urdu Tube (RUTUBE), where each comment is annotated as either abusive or non-abusive. **The third** dataset, which incorporates the Urdu script rather than Roman Urdu, is derived from the work documented by [Arif et al. \(2024\)](#), previously utilized by [Akhter et al. \(2020\)](#); [Amjad et al. \(2022\)](#), cate-

<sup>1</sup><https://github.com/shaheerakr/roman-urdu-abusive-comment-detector/blob/master/dataset.csv>

gorizing it with binary labels as either abusive or non-abusive. We name this dataset the Urdu Script Abuse Detection Corpus (USADC).

**Tasks** For RUHSOLD, we conduct coarse- and fine-grained evaluations using both zero-shot and few-shot settings across English, Roman Urdu, and Urdu prompts. The zero-shot prompts are shown in Table 7, and the few-shot prompts in Table 10. For RUTUBE and USADC, we restrict our study to English-based prompts. This decision is motivated by: (i) Roman Urdu and Urdu prompts consistently demonstrated lower performance across all LLMs compared to English, and (ii) the extensive experimentation conducted on RUHSOLD demonstrates sufficient generalizability across all language prompts. The same English prompt utilized for the coarse-grained (binary) task in RUHSOLD is used for RUTUBE and USADC. In few-shot (K-SHOTS) settings, we follow the same procedure as zero-shot but augment prompts with dataset-specific examples. For all datasets, we adopt a 4-shot setup (two abusive and two non-abusive examples), except for the fine-grained RUHSOLD task, where a 5-shot setup is used with one example per class (normal, abusive, sexual hate, religious hate, profane). This design ensures balanced exposure across categories despite label distribution variations, thus reflecting consistency with real-world settings. Furthermore, we manually curate all K-SHOTS examples from each dataset to accurately represent each class.

**Models** We evaluate pre-trained LLMs without fine-tuning, using zero-shot and few-shot prompting techniques to assess in-context learning across standard benchmarks. Our framework includes open-source models such as Llama-3 (1B-70B) (Dubey et al., 2024; Meta, 2024), Qwen-2 (7B, 72B) (Yang et al., 2024), GPT-4, and the Urdu-trained Lughaat, alongside reimplemented fine-tuned baselines for fair comparison. Our code is available at the link<sup>2</sup>. Please refer to Appendix A for details on the hardware, models, technical specifications, and answer parsing.

## 5 Results & Discussion

**Coarse- & Fine-Grained Task** In our first set of experiments, we use English, Roman Urdu, and Urdu prompts on the coarse- and fine-grained RUH-

<sup>2</sup><https://github.com/nii-yamagishilab/offensive-urdu-language-detection>

Model	COARSE-GRAINED					
	English		Roman Ur.		Urdu	
	Acc.	F1	Acc.	F1	Acc.	F1
Llama-3.2-1B	57.76	54.47	51.87	57.50	53.43	45.43
+ K-SHOTS	<b>60.20</b>	<b>60.20</b>	52.90	58.80	52.19	48.63
Llama-3.2-3B	63.60	64.08	55.93	50.87	54.67	48.60
+ K-SHOTS	<b>66.47</b>	<b>67.12</b>	56.25	53.53	56.77	49.55
Llama-3-8B	65.22	65.30	63.88	63.88	55.79	44.69
+ K-SHOTS	<b>70.88</b>	<b>69.82</b>	66.66	65.02	59.12	49.79
Llama-3-70B	74.36	74.36	70.26	70.76	70.83	68.88
+ K-SHOTS	<b>78.41</b>	<b>78.41</b>	73.77	73.11	73.68	71.07
Qwen-2-7B	73.21	73.21	69.17	67.40	-	-
+ K-SHOTS	<b>76.90</b>	<b>75.77</b>	70.03	69.80	-	-
Qwen-2-72B	82.07	81.11	72.32	75.91	-	-
+ K-SHOTS	<b>84.53</b>	<b>85.75</b>	74.55	76.40	-	-
Lughaat-1-8B	57.76	49.96	52.35	50.85	49.84	51.18
+ K-SHOTS	<b>61.58</b>	<b>58.75</b>	54.09	45.49	49.28	51.40
GPT-4	91.38	92.17	84.46	84.46	88.78	87.78
+ K-SHOTS	<b>92.91</b>	<b>92.58</b>	86.05	85.37	90.14	89.05
<b>Supervised</b>						
BERT-M <sup>⚡</sup>	54.20	-	-	-	-	-
DISTIL-BERT <sup>⚡</sup>	52.80	-	-	-	-	-
XLM-R <sup>⚡</sup>	57.20	-	-	-	-	-
XLM-Roberta+CNN <sup>♣</sup>	88.00	88.00	87.21*	87.14*	-	-
RomUrEm+CNN <sup>♣</sup>	89.00	89.00	61.90*	60.92*	-	-
BERT+CNN <sup>♣</sup>	90.00	90.00	<b>89.21*</b>	<b>89.17*</b>	-	-
FINE-GRAINED						
Llama-3-8B	59.11	53.33	21.27	20.95	15.11	15.38
+ K-SHOTS	<b>58.15</b>	<b>53.15</b>	48.55	47.08	31.77	31.70
Llama-3-70B	67.34	62.04	62.79	60.48	57.59	55.36
+ K-SHOTS	<b>72.67</b>	<b>69.25</b>	69.21	68.86	68.34	68.33
Qwen-2-7B	52.16	49.90	50.34	51.60	-	-
+ K-SHOTS	<b>53.11</b>	<b>49.63</b>	58.90	59.15	-	-
Qwen-2-72B	73.96	68.31	59.71	59.39	-	-
+ K-SHOTS	<b>75.77</b>	<b>70.30</b>	62.20	60.89	-	-
Lughaat-1-8B	44.45	44.50	29.77	29.80	<b>49.59</b>	<b>49.81</b>
+ K-SHOTS	29.59	32.81	22.23	22.81	42.84	43.47
GPT-4	75.50	75.02	68.32	68.28	63.27	62.98
+ K-SHOTS	<b>78.84</b>	<b>77.56</b>	70.52	69.90	68.24	65.09
<b>Supervised</b>						
XLM-ROBERTa+CNN <sup>♣</sup>	81.00	72.00	<b>79.57*</b>	69.39*	-	-
RomUrEm+CNN <sup>♣</sup>	75.00	64.00	55.92*	51.27*	-	-
BERT+CNN <sup>♣</sup>	82.00	75.00	<u>78.47*</u>	<b>75.83*</b>	-	-

Table 2: Summary of RUHSOLD results for coarse- and fine-grained tasks under zero- and few-shot settings across English, Roman Urdu, and Urdu prompts. Qwen-2 variants cannot process Urdu prompts. Bold values denote each model’s best performance across tasks and metrics. References include models marked with (⚡) from Ullah et al. (2023) and (♣) from Rizwan et al. (2020) where results marked with (\*) indicate fine-tuned models that we reimplemented for consistent evaluation.

SOLD tasks using state-of-the-art LLMs as shown in Table 2. Pilot experiments showed optimal performance with 4-shot examples (Figure 3), leading us to consider this setup for subsequent prompts. It can be seen in **coarse-grained task**, 4-shot (K-

SHOT) learning results in clear gains over zero-shot, with higher F1 scores and accuracy in distinguishing abusive from non-abusive tweets. It can also be seen that the prompts in Roman Urdu achieved performance comparable to those in English, especially when applied to large-size models such as Llama-3-70B-Instruct, Qwen-2-72B-Instruct, and GPT-4. We think that the difficulty smaller language models (Llama-1B, -3B) face with long prompts (K-SHOTS) arises from their limited ability to process extended input, despite its high relevance to the task. Consequently, the performance gap of smaller size LLMs between zero-shot and K-SHOT settings remains relatively small. The Lughaat-8B model, developed for Urdu, shows comparatively lower performance on coarse-grained classification tasks than models of a similar scale, such as Llama-8B and Qwen-7B, when evaluated on English and Roman Urdu prompts. Furthermore, the model exhibited notable difficulties in K-SHOT settings for both Roman Urdu and Urdu prompts, highlighting its limited ability to process longer inputs. Documentation and prior usage of this model are limited; however, given its status as one of the few publicly available Urdu-focused models, we include its results for completeness (see Section C).

In Roman Urdu prompt, Qwen-72B outperformed Llama-70B with F1-score of 76.40 against 73.11 and similarly, Qwen-7B outshone Llama-8B scoring 69.80 against 65.02, respectively, using K-SHOTS, despite both pairs having comparable model sizes. Due to the fact that Alibaba's<sup>3</sup> (Yuan and Zhang, 2024; Khaskheli and Jun, 2016) Qwen-2 model supports 29 languages, including key South Asian languages, such as Urdu, Hindi, and Bengali, it has demonstrated exceptional adept in processing Urdu as a result of its extensive multilingual training. Our findings align well with Qwen-2's superior performance on Roman Urdu prompts compared to similarly sized Llama-3 models, particularly in binary abusive detection task. However, it can also be seen that the Qwen-2 variants completely failed to process Urdu prompts, while the Llama-3 variants performed better in understanding Urdu language prompts. The use of Urdu prompts overall provides the lowest quality results. This is consistent with the findings by Nguyen et al. (2025), who report that non-Latin script languages, such as Urdu, continue to underperform despite strong multilingual benchmark scores. In partic-

<sup>3</sup><https://www.alibaba.com/>

ular, Urdu, an abjad-derived script adapted from Arabic and Persian is based on intricate ligatures, contextual character forms, and joining rules, all of which pose challenges for tokenization and representation (Fatima et al., 2007; Zia et al., 2018). Nevertheless, our results show that larger models, such as Llama-70B still performed reasonably well in Urdu under K-SHOT settings, achieving results comparable to those for Roman Urdu, suggesting that increased model size can partially mitigate this limitation. This highlights the adequacy of open source Llama-70B to effectively process Roman Urdu and Urdu, demonstrating its strong generalization across linguistic variations, as evidenced by the findings of our study.

With the English prompt, GPT-4 substantially outperformed the others in 4-shot scenarios achieving an F1-score of 92.58, compared to 92.17 in the regular (zero-shot) setup. For Qwen-2-72B, the model consistently shows a robust performance outperforming Llama-3-70B, achieving an F1 score of 81.11 against 74.36, even without k-shot-augmented examples. In general, among the models, it is observed that GPT-4 consistently achieves superior performance in all the prompt-based methodologies tested, including those in English, Roman Urdu and Urdu. Overall, in the coarse-grained task, all models achieve their highest performance with English prompts, show comparable results in Roman Urdu, and the lowest in Urdu. This limitation stems from the fact that most LLMs are primarily optimized for Latin scripts, reducing their effectiveness on morphologically complex scripts such as Urdu. In contrast, English prompts align with model's dominant training language, use familiar instruction patterns, reducing ambiguity and tokenization noise that arise in Urdu or Roman Urdu (see Appendix C for further details on prompt language effects).

We also compare our approach against state-of-the-art fine-tuned models, noting that, to the best of our knowledge, no previous studies have employed pretrained LLMs on RUHSOLD. For this study, we reimplemented the fine-tuned models from prior work. However, since these models rely on fine-tuning rather than in-context learning, direct comparisons should be made with caution. Specifically, we reimplemented the fine-tuned models XLM-RoBERTa+CNN, RomUrEm+CNN, and BERT+CNN (Rizwan et al., 2020), and included their original results alongside ours marked with an

asterisk (\*) in Table 2, to ensure consistency. While these models achieve strong performance, it is important to note that they were fine-tuned on 4.7M randomly collected hate-speech tweets (Rizwan et al., 2020), whereas no fine-tuning is performed in our settings. It can be seen that the GPT-4 K-SHOT outperforms all baselines with an F1-score of 92.58, followed by the fine-tuned BERT+CNN model at 89.17. Notably, Qwen-72B attains a comparable score of 85.75 without any fine-tuning. For completeness, we also report the corresponding precision and recall results for Table 2 in Table 5.

Our overarching aim in this study is to determine whether LLMs can process more nuanced categories of offensive language. To this end, we repeat our experiments on the **fine-grained** RUSHSOLD, results summarized in the lower section of Table 2. In this task, the 'Abusive' category is further divided into four sublabels, as shown in Table 1. The zero-shot prompts are listed in Table 7, and the K-SHOT context examples are provided in Table 10 in the Appendix. Overall, we observe similar trends to those found in the coarse-grained task, with a few notable differences. Llama-8B exhibits strong performance in English prompts, while showing larger performance gaps between K-SHOT and zero-shot settings in Roman Urdu and Urdu. This suggests that few-shot examples in Roman Urdu and Urdu enhance cross-script generalization, while zero-shot comprehension of five-class labels remains challenging. Details on contributing factors to this unusual behavior of Llama-8B are provided in the Appendix B. Interestingly, Qwen-72B K-SHOTS achieve accuracy in English prompts comparable to GPT-4 (zero-shot), underscoring its robust fine-grained classification capabilities of hate speech. For Lughaat-8B, we find strong performance in Urdu prompts compared to Roman Urdu (F1 of 49.81 vs. 29.80 in zero-shot), reflecting its pre-training on native Urdu script. However, the performance declines with K-SHOT in all prompts for this model, indicating a limited ability to process longer contextual input.

Except for Lughaat, all other models show improved performance with English prompts, with GPT-4 K-SHOT achieving the highest F1 score of 77.56, followed by Qwen-72B with 70.30. While in Urdu prompts, the Llama-70B K-SHOT model outperforms GPT-4, achieving an F1 score of 68.33 compared to 65.09, and shows nearly equivalent performance in Roman Urdu prompts, demonstrat-

Model	RUTUBE		USADC	
	Acc.	F1	Acc.	F1
Llama-3-8B	66.99	65.95	50.83	47.80
+ K-SHOTS	66.48	68.12	69.08	68.70
Llama-3-70B	72.70	72.07	73.83	72.83
+ K-SHOTS	76.00	74.55	79.90	78.51
Qwen-2-7B	69.01	66.20	67.76	66.97
+ K-SHOTS	69.35	68.60	72.88	72.45
Qwen-2-72B	79.99	79.20	79.34	78.89
+ K-SHOTS	<u>83.89</u>	<u>82.46</u>	<u>81.24</u>	<u>81.24</u>
Lughaat-1-8B	52.59	11.56	52.34	47.82
+ K-SHOTS	78.34	75.36	56.79	55.20
GPT-4	86.70	85.95	86.41	86.34
+ K-SHOTS	<b>87.33</b>	<b>86.15</b>	<b>88.85</b>	<b>88.78</b>
<b>Supervised</b>				
SimpleLogistic Reg.*	85.50*	85.61*	74.56*	77.86*
LogitBoost Reg.*	85.40*	85.35*	75.43*	77.77*
XLM-Roberta+CNN♣	<u>90.15*</u>	<u>90.11*</u>	<b>84.21*</b>	<b>84.18*</b>
RomUrEm+CNN♣	68.60*	68.46*	70.17*	69.16*
BERT+CNN♣	<b>92.55*</b>	<b>92.53*</b>	<u>78.94*</u>	<u>77.97*</u>
<b>No Fine-tuning</b>				
Llama-3-8b♣ (0-shot)	-	-	-	44.73
Llama-3-8b♣ (6-shot)	-	-	-	71.64
GPT-4♣ (0-shot)	-	-	-	86.27
GPT-4♣ (6-shot)	-	-	-	88.71

Table 3: Comparison of RUTUBE and USADC to evaluate LLMs generalization capabilities in terms of accuracy and F1 scores. References include models marked with (\*) from Akhter et al. (2020) and (♣) from Arif et al. (2024). Best performing results across all datasets, approach and metric are highlighted in bold and underlined. Results marked with (\*) indicate fine-tuned models that we reimplemented for consistent evaluation.

ing its robust multilingual capability and adaptability to diverse language instructions. Although fine-tuned baselines remain competitive, open-source LLMs achieve comparable results without any training. Among the fine-tuned approaches, GPT-4 + K-SHOTS achieves the highest F1 score (77.56), followed by BERT+CNN (75.83), Qwen-72B (70.30), and XLM-RoBERTa+CNN (69.39). These findings highlight the potential of instruction-tuned LLMs to rival fine-tuned baselines, even in complex multi-classification tasks.

**RUTUBE versus USADC** In addition to RUSHSOLD, we evaluate pretrained LLMs performance on RUTUBE and USADC using zero- and few-shot methods. In this experiment, we also report the results of reimplemented models, denoted with (\*). For fair comparison, we report only the results obtained using English prompts against fine-tuned

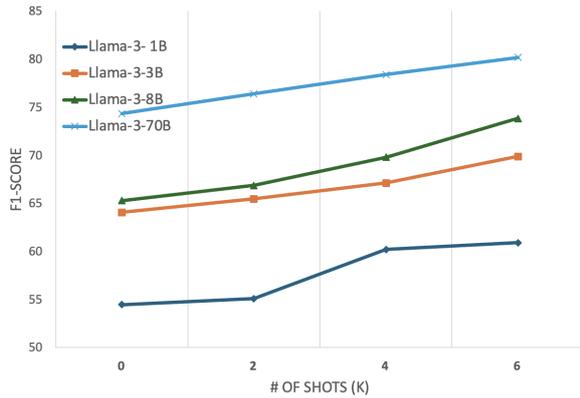


Figure 1: Influence of ‘k’ in few-shot setting with English prompts for zero-, 2-, 4-, and 6-shot experiments on Llama-3.2 and Llama-3 models of various sizes.

baselines. Prior studies have evaluated the RUTUBE dataset only in (Akhter et al., 2020) and the USADC dataset in (Arif et al., 2024). Therefore, we reproduce experiments using fine-tuned approaches from Rizwan et al. (2020) on both datasets for consistency (see Appendix A for model details). Since, Akhter et al. (2020) reported results for fine-tuned approaches, such as Simple Logistic Regression and LogitBoost Regression—trained on entire 147,000 instances of the RUTUBE dataset, we use a sampled subset of 10,000 instances for fair comparison with pretrained LLMs.

Our findings, summarized in Table 3, show that despite the absence of model fine-tuning, large-scale models such as Qwen-72B achieve competitive performance. Among the in-context learning approaches, GPT-4 and Qwen-72B emerge as the top-performing models across both RUTUBE and USADC. Overall, fine-tuned models exhibit superior performance on the RUTUBE, which is written in Roman Urdu, indicating that fine-tuned approaches still hold a performance edge in handling the Roman Urdu. However, on the USADC (composed in Urdu script), GPT-4 achieves the highest F1 score (88.78), followed by XLM-RoBERTa+CNN (84.18), Qwen-72B (81.24), Llama-70B (78.51), and BERT+CNN (77.97). These results highlight that while fine-tuned models perform well on Roman Urdu data, datasets written in Urdu script remain challenging, where in-context learning approaches without model fine-tuning demonstrate relatively stronger generalization.

**Role of ‘k’** We also explore the impact of k-shot learning by testing 1, 2, 4, and 6-shot configurations to determine their impact on the model perfor-

mance. Specifically, we choose the Llama-3 model variants across varying sizes (1B, 3B, 8B, and 70B) to evaluate the influence of different shot examples in a binary coarse-grained task on the RUHSOLD dataset. Each k-shot scenario is designed to evenly distribute examples between the abusive and non-abusive categories. For example, in a 2-shot setup, one example is drawn from abusive content and another from non-abusive content. As shown in Figure 3, a progressive increase can be seen in the F1-score for binary classification as the number of shots increases, highlighting the benefit of more examples. However, the incremental performance gains tend to show no significant increase beyond 4-shots, when advancing to 6-shots. These results indicate that while the addition of more examples initially increases the model’s ability to discriminate between abusive and non-abusive text, there is a threshold beyond which additional examples lead to no extra gains.

## 5.1 Error Analysis

To gain a deeper understanding of the zero-shot abilities of LLMs in our tasks, we performed a qualitative analysis of the predictions made by Llama-3-8B-Instruct, Llama-3-70B-Instruct, Qwen-2-7B-Instruct, and Qwen-2-72B-Instruct on the **coarse-grained** RUHSOLD dataset. Detailed results from this analysis are presented in the Appendix under Tables 11, 12, 13, and 14, respectively. These tables include examples of both correct and incorrect predictions, along with insights into the models’ reasoning processes, illustrating why certain deviations from expected performance occurred. We noted that in examples where the model identifies an even slight derogatory element, the sample is predicted as abusive. To this end, we also provide a detailed analysis of the fine-grained classification behavior of Llama-3-8B-Instruct and Qwen-2-7B-Instruct on RUHSOLD, revealing that both models tend to overgeneralize into the “Abusive” class, frequently misclassifying non-toxic content as abusive. A major limitation of Llama-8B is its poor performance on the Sexual hate category, where only a small number of instances are correctly recognized, as shown in Figure 2. This occurs despite these instances being labeled as non-abusive in the gold-standard annotations (Appendix C).

Table 15 in the Appendix presents illustrative examples of **fine-grained** classification outputs of RUHSOLD across different hate speech categories,

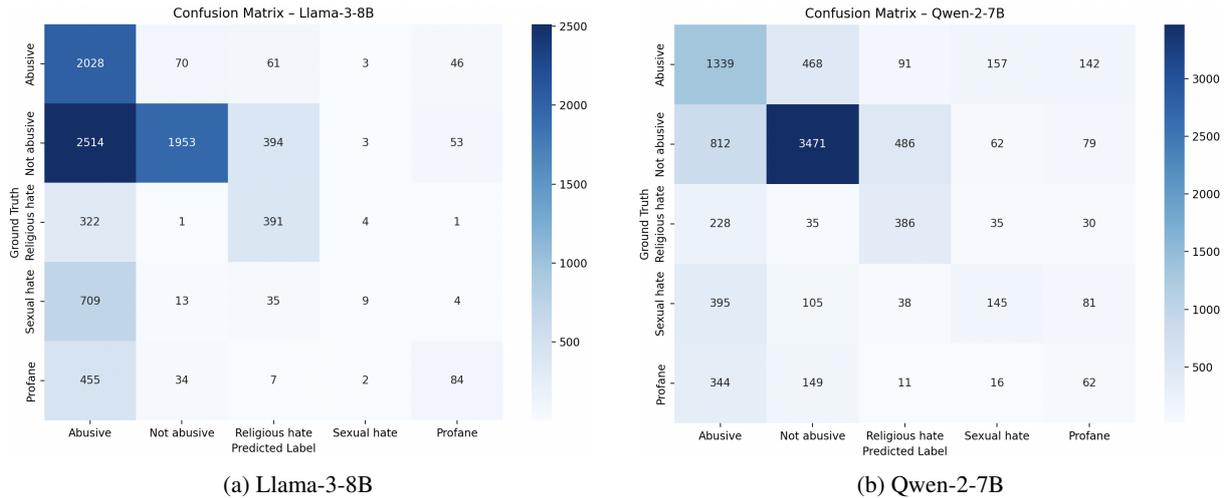


Figure 2: Comparison of model performance across coarse-grained RUHSOLD using Roman Urdu prompt with few-shot settings for (a) Llama-3-8B and (b) Qwen-2-7B

Model	Responsiveness to prompts provided in		Given English prompt can process	
	Roman Urdu	Urdu	Roman Urdu	Urdu
Llama-3	✓	✓	✓	✓
Qwen-2	✓	✗	✓	✓
FLAN-T5	Partial	✗	✓	Partial
Falcon	Partial	✗	✓	✓
GPT-4	✓	✓	✓	✓

Table 4: Comparative analysis of various LLMs responsiveness to prompts provided in Roman Urdu and Urdu language, while also detailing each model’s ability to interpret and process prompts provided in English across these languages.

highlighting the challenges LLMs face when making distinctions at a granular level. Our analysis indicates that the Lughat model often confuses sexual hate with religious hate, misclassifying instances of the former as the latter. In contrast, LLaMA and Qwen models exhibit greater confusion between abusive and profane language. These patterns show the limitations that current LLMs face in capturing the nuanced boundaries of human language, particularly between profanity and targeted abusive expressions. Furthermore, our study also explores the selection of LLMs based on their ability to process prompts in the Urdu language as part of pilot experiments. Table 4 provides a summary of various LLMs in handling Roman Urdu and Urdu. In addition, since our datasets span both Roman Urdu (RUHSOLD, RUTUBE) and Urdu script (USADC), we conduct a **cross-linguistic dataset evaluation**, detailed in Appendix D. Appendix E provide further information on the severity of toxi-

city across the datasets. In addition, supplementary experiments demonstrate the results of a supervised approach applied to the Roman Urdu datasets.

## 6 Conclusion

This paper aims at advancing our understanding of abusive language detection in linguistically challenging scenarios for the under-resourced Urdu language, a task of significant importance in social media, news media, and the research community. With the increase in digital media usage, there is a growing need for automated text classification systems that can address offensive content in non-English languages without requiring extensive training. Our extensive experimental setup underscores the potential and limitations of applying LLMs in a zero-shot and few-shot configuration to classify sentences that manifest abuse. In addition to exploring the ability of LLMs to detect abusive content, we also examine whether prompts written in non-English languages like Roman Urdu and Urdu provide enough information to effectively distinguish between abusive and non-abusive content. While English prompts continue to deliver the highest performance, Roman Urdu prompts also significantly improve the quality of prompt engineering; however, prompts in Urdu tend to result in lower quality prompts more frequently. We hope our work inspires further research on offensive language detection in under-resourced settings. Future work may extend to more aspects of the Urdu language, such as cultural, racial, or age-related content, to better understand the capabilities and limitations of LLMs.

## Limitations

The field of offensive language detection in Urdu utilizing in-context learning remains an under-explored area. Hence, there exists many potential research directions to explore. Variability in the sizes of datasets and their labels presents multiple opportunities for future research. Moreover, datasets on hate speech in non-English languages are notably limited. Our investigation has primarily focused on the Urdu language and its representation in Roman Urdu. Moreover, the baseline models used in comparative study are not in direct competition with the models evaluated in this research. Future studies may involve more LLMs to broaden the scope of in-context learning to include additional languages, further exploring the capabilities and limitations of LLMs.

## Ethical Considerations

The task of detecting hate speech in digital media is a persistent and critical challenge, which makes it crucial to have fast and dependable methods to curb the dissemination of harmful content. Traditional approaches to hate speech detection often depend on extensive model training, which can restrict their applicability to real-world scenarios. Our methodology promotes enhanced flexibility, especially through the use of large language models. Nevertheless, it is imperative to strike a judicious balance that ensures accurate predictions, model adaptability, and the preservation of individuals' rights to access truthful information.

Finally, we recognize a potential ethical concern in presenting examples of highly abusive content. Therefore, we urge readers to approach this material with caution. Our intention is to advance research and understanding of offensive language detection, not to promote or normalize such language. We also encourage researchers and practitioners to consider the sensitive nature of the content and the potential psychological impact on those analyzing it.

## Acknowledgements

The authors wish to express gratitude to the funding organization, as this work is supported by the JST CREST Grant (JPMJCR20D3), Japan, and the TSUBAME 4.0 supercomputer at the Institute of Science, Tokyo, whose computational resources are gratefully acknowledged.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Wafia Adouane, Nasredine Semmar, and Richard Johansson. 2016. **Romanized Berber and Romanized Arabic automatic language identification using machine learning**. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 53–61, Osaka, Japan. The COLING 2016 Organizing Committee.
- Muhammad Pervez Akhter, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed Abdelmajeed, and Muhammad Tariq Sadiq. 2020. Automatic detection of offensive language for urdu and roman urdu. *IEEE Access*, 8:91213–91226.
- Jesujoba O Alabi, Israel Abebe Azime, Miaoran Zhang, Cristina España-Bonet, Rachel Bawden, Dawei Zhu, David Ifeoluwa Adelani, Clement Oyeleke Odoje, Idris Akinade, Iffat Maab, and 1 others. 2025. Afridoc-mt: Document-level mt corpus for african languages. *arXiv preprint arXiv:2501.06374*.
- Mehreen Alam and Sibte Ul Hussain. 2022. Roman-urdu-parl: Roman-urdu and urdu parallel corpus for urdu language understanding. *Transactions on Asian and Low-Resource Language Information Processing*, 21(1):1–20.
- Maaz Amjad, Alisa Zhila, Grigori Sidorov, Andrey Labunets, Sabur Butta, Hamza Imam Amjad, Oxana Vitman, and Alexander Gelbukh. 2022. Overview of abusive and threatening language detection in urdu at fire 2021. *arXiv preprint arXiv:2207.06710*.
- Bo An. 2023. Prompt-based for low-resource tibetan text classification. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(8):1–13.
- Zahra Ansari, Shaikat Ali, and Fatima Khan. 2020. Use of roman script for writing urdu language. *International Journal of Linguistics and Culture*, 1(2):165–178.
- Samee Arif, Abdul Hameed Azeemi, Agha Ali Raza, and Awais Athar. 2024. Generalists vs. specialists: Evaluating large language models for urdu. *arXiv preprint arXiv:2407.04459*.
- Elena Bashir. 2011. Urdu and linguistics: A fraught but evolving relationship.
- Iz Beltagy, Arman Cohan, Robert Logan IV, Sewon Min, and Sameer Singh. 2022. Zero-and few-shot nlp with pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 32–37.
- Steven Bird. 2001. Orthography and identity in cameroon. *Written Language & Literacy*, 4(2):131–162.

- Damian Blasi, Antonios Anastasopoulos, and Graham Neubig. 2021. Systematic inequalities in language technology performance across the world’s languages. *arXiv preprint arXiv:2110.06733*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Mahima Choudhary, Basant Agarwal, and Vishnu Goyal. 2025. Hate speech detection: Leveraging llm-gpt2 with fine-tuning and multi-shot techniques. *Procedia Computer Science*, 258:2817–2825.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. **Unsupervised cross-lingual representation learning at scale**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Thomas Davidson, Dana Warmley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Hugging Face. 2021. The ai community building the future. URL: <https://huggingface.co>.
- Tayyaba Fatima, R Islam, and M Anwar. 2007. Morphological and orthographic challenges in urdu language processing: A review. In *Proc. 13th Workshop Asian Lang. Resour.*, 1, pages 44–51.
- Paula Fortuna and Sérgio Nunes. 2018. **A survey on automatic detection of hate speech in text**. *ACM Comput. Surv.*, 51(4).
- Lei Gao, Alexis Kuppersmith, and Ruihong Huang. 2017. **Recognizing explicit and implicit hate speech using a weakly supervised two-path bootstrapping approach**. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 774–782, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. **Making pre-trained language models better few-shot learners**. In *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 3816–3830, Online. Association for Computational Linguistics.
- Maria Isabel Maldonado Garcia. 2011. The urdu language reforms. *Studies*, 26:97.
- Adrián Girón, Javier Huertas-Tato, and David Camacho. 2025. Llm synthetic generation to enhance online content moderation generalization in hate speech scenarios. *Computing*, 107(8):164.
- Lawrence Han and Hao Tang. 2022. Designing of prompts for hate speech recognition with in-context learning. In *2022 international conference on computational science and computational intelligence (csci)*, pages 319–320. IEEE.
- Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal romanization tool uroman. In *Proceedings of ACL 2018, system demonstrations*, pages 13–18.
- William Huang, Haokun Liu, and Samuel R Bowman. 2020. Counterfactually-augmented snli training data does not yield better generalization than unaugmented data. *arXiv preprint arXiv:2010.04762*.
- J Jaavid, Raj Dabre, M Aswanth, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15593–15615.
- Kelly Jackson. 2011. **Derald wing sue, microaggressions in everyday life: Race, gender, and sexual orientation**. *Social Service Review*, 85:519–521.
- Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. 2022. **A good prompt is worth millions of parameters: Low-resource prompt-based learning for vision-language models**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2763–2775, Dublin, Ireland. Association for Computational Linguistics.
- Yiping Jin, Leo Wanner, and Alexander Shvets. 2024. Gpt-hatecheck: Can llms write better functional tests for hate speech detection? *arXiv preprint arXiv:2402.15238*.
- Asadullah Khaskheli and Yun Jun. 2016. A review on the importance of e-commerce for smes in pakistan. *RISUS-Journal on Innovation and Sustainability*, 7(1-2016).
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Irene Kwok and Yuzhou Wang. 2013. **Locate the hate: Detecting tweets against blacks**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1):1621–1622.

- Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2021. Good examples make a faster learner: Simple demonstration-based learning for low-resource ner. *arXiv preprint arXiv:2110.08454*.
- Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, pages 14–17.
- AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Hoang H Nguyen, Khyati Mahajan, Vikas Yadav, Julian Salazar, Philip S. Yu, Masoud Hashemi, and Rishabh Maheshwary. 2025. Prompting with phonemes: Enhancing LLMs’ multilinguality for non-Latin script languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11975–11994, Albuquerque, New Mexico. Association for Computational Linguistics.
- Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. 2022. Cross-lingual retrieval augmented prompt for low-resource languages. *arXiv preprint arXiv:2212.09651*.
- Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, WWW ’16*, page 145–153, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Alexandra O’Neil, Daniel Swanson, Robert Pugh, Francis Tyers, and Emmanuel Ngué Um. 2023. Comparing methods of orthographic conversion for bàsàà, a language of cameroon. In *Proceedings of the Fourth workshop on Resources for African Indigenous Languages (RAIL 2023)*, pages 97–105.
- Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. 2024. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*.
- Kirk R Person. 2009. Heritage scripts, technical transcriptions, and practical orthographies: a middle path towards educational excellence and cultural preservation for thailand’s ethnic minority languages’. In *Proceedings from the international conference on national language policy: Language diversity for national unity*, pages 189–200.
- Flor Miriam Plaza-del Arco, M Dolores Molina-González, L Alfonso Urena-López, and M Teresa Martín-Valdivia. 2021. Comparing pre-trained language models for spanish hate speech detection. *Expert Systems with Applications*, 166:114120.
- Tariq Rahman. 1997. The urdu—english controversy in pakistan. *Modern Asian Studies*, 31(1):177–207.
- Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in roman urdu. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 2512–2522.
- Hafiz Hassaan Saeed, Muhammad Haseeb Ashraf, Faisal Kamiran, Asim Karim, and Toon Calders. 2021. Roman urdu toxic comment classification. *Language Resources and Evaluation*, pages 1–26.
- Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.
- Nathan Schucher, Siva Reddy, and Harm de Vries. 2022. The power of prompt tuning for low-resource semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 148–156, Dublin, Ireland. Association for Computational Linguistics.
- Jaehyung Seo, Hyeonseok Moon, Chanhee Lee, Sugyeong Eo, Chanjun Park, Jihoon Kim, Changwoo Chun, and Heuiseok Lim. 2022. Plain template insertion: korean-prompt-based engineering for few-shot learners. *IEEE Access*, 10:107587–107597.
- Muhammad Haroon Shakeel and Asim Karim. 2020. Adapting deep learning for sentiment classification of code-switched informal short text. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing, SAC ’20*, page 903–906, New York, NY, USA. Association for Computing Machinery.
- Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Auto-prompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.
- Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. Indicgenbench: A multilingual benchmark to evaluate generation capabilities of llms on indic languages. *arXiv preprint arXiv:2404.16816*.
- Chengyu Song, Taihua Shao, Kejing Lin, Dengfeng Liu, Siyuan Wang, and Honghui Chen. 2022. Investigating prompt learning for chinese few-shot text classification with pre-trained language models. *Applied Sciences*, 12(21):11117.
- Yirong Sun, Dawei Zhu, Yanjun Chen, Erjia Xiao, Xinghao Chen, and Xiaoyu Shen. 2024. Fine-grained and multi-dimensional metrics for document-level machine translation. *arXiv preprint arXiv:2410.20941*.

Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2024. Benchmarking pre-trained large language models’ potential across urdu nlp tasks. *arXiv preprint arXiv:2405.15453*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, and Marie-Anne Lachaux. 2023. Timothé e lacroix, baptiste roziè re, naman goyal, eric hambro, faisal azhar, auré lien rodriguez, armand joulain, edouard grave, and guillaume lamplé. 2023 a. llama: Open and efficient foundation language models. corr, vol. abs/2302.13971 (2023). *LLaMA: Open and Efficient Foundation Language Models. CoRR, Vol. abs/2302.13971*.

Kurban Ubul, Gulzira Tursun, Alimjan Aysa, Donato Impedovo, Giuseppe Pirlo, and Tuergen Yibulayin. 2017. Script identification of multi-script documents: a survey. *IEEE access*, 5:6546–6559.

Faizad Ullah, Ubaid Azam, Ali Faheem, Faisal Kamiran, and Asim Karim. 2023. Comparing prompt-based and standard fine-tuning for urdu text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6747–6754.

Betty Van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. [PromDA: Prompt-based data augmentation for low-resource NLU tasks](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Zeeraq Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Zeeraq Waseem 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.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. *arXiv preprint arXiv:2109.07684*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Elaine J Yuan and Lin Zhang. 2024. The political economic process of ‘platformization’: The historical trajectory of alibaba. *The Economic Lives of Platforms: Rethinking the Political Economy of Digital Markets*, page 108.

Haris Bin Zia, Agha Ali Raza, and Awais Athar. 2018. Urdu word segmentation using conditional random fields (crfs). *arXiv preprint arXiv:1806.05432*.

## A Technical Specifications

**Hardware** We utilize two types of GPUs: the 16-GB NVIDIA V100 and the 40-GB NVIDIA A100. These GPUs are accessed through nodes in a large cluster, and each node is equipped with multiple GPUs. To enhance the speed of inference, some of our experiments leverage data parallelism.

### Pretrained Models & Implementation Specifics

We used PyTorch to implement the models, borrowing from HuggingFace (Face, 2021) for Llama-3 Meta AI’s third-generation LLM family (Meta, 2024; Dubey et al., 2024) instruction tuned variants including Llama-3.2-1B-Instruct<sup>4</sup>, Llama-3.2-3B-Instruct<sup>5</sup>, Llama-3-8B-Instruct<sup>6</sup>, and Llama-3-70B-Instruct<sup>7</sup>. In addition, we use Alibaba’s advaced LLM series Qwen-2 (Yang et al., 2024) instruction tuned models including Qwen-2-7B-Instruct<sup>8</sup>, and Qwen-2-72B-Instruct<sup>9</sup>, spanning a broad spectrum of computational capacities. We also use an Urdu language model trained on lughaat-urdu-dataset-llm named Lughaat-1.0-8B-Instruct<sup>10</sup>. Another relevant model is Alif-1.0-8B-Instruct, designed for Urdu and English, however, due to missing configuration files, we were unable to integrate it into our experiments in the current work. Lughaat is built on the Llama-3.1-8B architecture and has been reported to outperform Alif-8B. Limited documentation or prior research exists on Lughaat; however, as it is the only publicly available Urdu-based model, we included its results in our evaluation for completeness. Finally, for GPT-4, we utilized the official OpenAI API and tested it at various temperature settings (0, 0.5, and 0.8). Despite these variations, we noted minimal differences in the generated texts. We report accuracy and macro F1 scores.

**Fine-tuned Approaches** We also compare our approach with state-of-the-art fine-tuned models for offensive language detection in Urdu by reim-

<sup>4</sup><https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct>

<sup>5</sup><https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

<sup>6</sup><https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct>

<sup>7</sup><https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct>

<sup>8</sup><https://huggingface.co/Qwen/Qwen2-7B-Instruct>

<sup>9</sup><https://huggingface.co/Qwen/Qwen2-72B-Instruct>

<sup>10</sup><https://huggingface.co/muhammadnoman76/Lughaat-1.0-8B-Instruct>

plementing the architectures introduced in (Rizwan et al., 2020). Specifically, we use pretrained embeddings from BERT (Devlin et al., 2018), and XLM-RoBERTa (Conneau et al., 2020), a multilingual variant of RoBERTa trained on 100 languages—and fine-tune them on the RUHSOLD dataset to evaluate their transfer learning capabilities across domains and languages. These models are referred to as BERT+CNN and XLM-RoBERTa+CNN, respectively. In addition, Rizwan et al. (2020) introduced domain-specific embeddings, termed RomUrEm, trained using word2vec (Mikolov et al., 2013) on a corpus of approximately 4.77 million Roman Urdu tweets, which were integrated into the RomUrEm+CNN model. All three models share the same CNN-gram architecture but differ in their source of embeddings. In our reimplementation, we adopt BERT+CNN and XLM-RoBERTa+CNN with their original pretrained embeddings, while RomUrEm+CNN is re-evaluated without using the pretrained domain-specific embeddings to assess its dependence on language-specific features. As a result, we observe a performance drop of approximately 15 to 20 F1 points across both coarse- and fine-grained RUHSOLD tasks (Table 2), confirming that the absence of pretrained Roman Urdu embeddings substantially weakens performance compared to the original setup and the in-context learning approaches. We also evaluate the performance of these models on the RUTUBE and USADC datasets (Table 3). Furthermore, since Akhter et al. (2020) evaluated fine-tuned models such as Simple Logistic Regression and LogitBoost Regression on the entire 147,000 instances of the RUTUBE dataset, we instead used a balanced subset of 10,000 samples from the same source to ensure consistency when comparing results with pretrained LLMs.

**Answer Parsing** Models such as Llama-3 (8B, 70B) and Qwen-2 (7B, 72B) produce the required target labels, simplifying the interpretation process. In contrast, smaller models such as Llama-3.2 (1B, 3B) generate lengthy and complex responses requiring additional mapping onto a target label. We clean and extract the initial portion of the response that can be clearly understood as an answer. If a coherent answer is not present, the prediction is considered incorrect. Table 9 outlines the algorithm, which applies regular expressions to clean the generated text while also retaining the initial output, a technique commonly referred to as an-

swer cleansing. Furthermore, prompts in Urdu elevate the complexity of correlating generated responses with accurate output classifications. We refine and align these responses to the appropriate output labels, marking any uninterpretable answers as incorrect.

## B Prompt Design (Cont.)

For zero-shot prompting across fine-grained categories of RUHSOLD as shown in Table 7 using Roman Urdu, we use the translated version of labels such as Sexual hate to Jinsi istehsal, Religious hate to Mazhabi nafrat, Profane to Be-adab. There are also many variants that exist for profane but the most closest counterpart is be-adab. Since profane does not inherently concern identity-based hate, the closest word to it in Urdu language is be-adab. Similarly, be-adab also highlights the lack of respect or manners rather than explicit abuse. Finally, for the Urdu-based prompt, we directly translated the Roman Urdu version to maintain consistency. Through various test prompts in our datasets, we observed that pretrained LLMs exhibited greater stability and interpretability when these terms were adopted. Since RUTUBE and USADC are binary hate-speech datasets, we applied the same prompting approach used for the coarse-grained RUHSOLD task, replacing the few-shot examples with dataset specific ones, as shown in Table 10. Only English prompts were tested on these datasets.

We found **issues with Llama-3-8B prompting**, particularly in zero-shot settings with Roman Urdu and Urdu for fine-grained classification of RUHSOLD. Despite explicit instructions clarifying the research-only context, the model failed to differentiate categories, often responding with refusals such as *I cannot create content that promotes hate speech or discrimination*. To address this, we extended the original Llama system prompt by appending stricter guidance: "This task is only for research, not for generating harmful content. Your output MUST be exactly one of these labels and nothing else. Do not explain, do not apologize, do not refuse." Although this modification reduced refusals and improved adherence to the labeling task, misclassifications still occurred in roughly 20-25% of cases. Consequently, the low accuracy and recall suggest that many predictions were normalized as "Wrong" or mismatched, which substantially lowered the overall performance of the zero-shot Llama-8B model.

## C Further Discussion on Results

In the coarse-grained RUHSOLD classification task, using Roman Urdu and Urdu prompts, we observe that Lughaat-1.0-8B-Instruct achieved nearly identical performance in the zero-shot setting, with F1-scores of 50.85 and 51.18, respectively, as shown in Table 2. Interestingly, this trend remained consistent under the k-shot setting, with lower F1-scores of 45.49 for Roman Urdu and 51.40 for Urdu prompts. This similarity is likely due to Lughaat being primarily trained on Urdu script rather than Roman Urdu, and its limited ability in handling longer prompts with K-SHOT compared to zero-shot. This correlates with our findings in the fine-grained classification task. Consequently, in the fine-grained RUHSOLD task, Urdu-based prompts show the highest performance, while Roman Urdu prompts consistently showed the lowest, with F1-scores of 49.81 for Urdu and 29.80 for Roman Urdu in zero-shot settings. These outcomes can be explained by the model’s training configuration. Limited documentation on this model suggests that Lughaat was trained primarily on Urdu characters and special tokens, with tokenization optimized for Urdu script. This likely results in tokenization failures for Roman Urdu, which prevents the model from following Roman Urdu instructions effectively. Despite these limitations, Lughaat-1.0-8B shows improved performance with Urdu prompts compared to its similarly sized Llama-3-8B model on coarse- and fine-grained RUHSOLD tasks. This highlights the potential of Urdu-specific LLMs and underscores the importance of further development of models tailored to underrepresented languages such as Urdu and Roman Urdu.

**Prompt Language Effects** English prompts perform best among all settings because most LLMs (e.g., GPT, Llama, Qwen) are predominantly trained and instruction-tuned in English, aligning their embeddings, syntax, and semantics closely with it. In contrast, Urdu prompts introduce tokenization inefficiencies due to its cursive Nastaliq script, an abjad derived from Persian and Arabic that omits short vowels and exhibits complex morphology, while Roman Urdu adds further ambiguity through inconsistent spellings (e.g., *acha*, *achha*, *achaah* for the English word okay). These characteristics fragment token representations and reduce semantic coherence in multilingual settings.

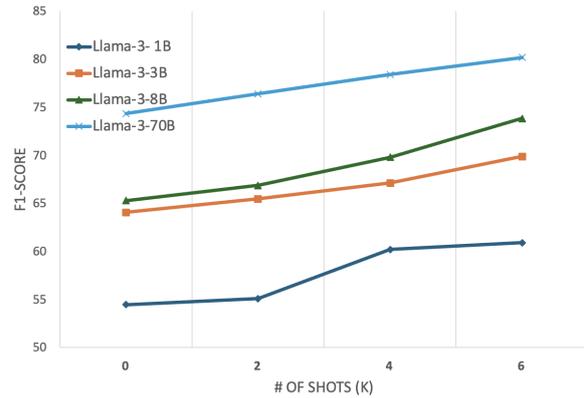


Figure 3: Influence of ‘k’ in few-shot setting with English prompts for zero-, 2-, 4-, and 6-shot experiments on Llama-3.2 and Llama-3 instruction models of various sizes.

**Role of ‘k’** We also explore the impact of k-shot learning by testing 1, 2, 4, and 6-shot configurations to determine their impact on the model performance. Specifically, we choose the Llama-3 model variants across varying sizes (1B, 3B, 8B, and 70B) to evaluate the influence of different shot examples in a binary coarse-grained task on the RUHSOLD dataset. Each k-shot scenario is designed to evenly distribute examples between the abusive and non-abusive categories. For example, in a 2-shot setup, one example is drawn from abusive content and another from non-abusive content. As shown in Figure 3, a progressive increase can be seen in the F1-score for binary classification as the number of shots increases, highlighting the benefit of more examples. However, the incremental performance gains tend to show no significant increase beyond 4-shots, when advancing to 6-shots. These results indicate that while the addition of more examples initially increases the model’s ability to discriminate between abusive and non-abusive text, there is a threshold beyond which additional examples lead to no extra gains.

**Error Analysis (Cont.)** We observe that for smaller models such as Llama-8B and Qwen-7B as illustrated in Figure 2, they exhibit notable differences in their fine-grained classification behavior on the RUHSOLD dataset. The strongest predictions for both models are observed in the "Not abusive" class. However, Llama-8B frequently misclassifies neutral comments as Abusive, reflecting its tendency to over-flag non-toxic content and achieve high recall but low precision. In contrast, Qwen-7B produces fewer such misclassifications, maintain-

ing higher recall with moderate precision. A key weakness of Llama-8B lies in its handling of the Sexual hate category, where only a few instances are correctly identified, and most are incorrectly classified as Abusive. The model also struggles to differentiate between profanity and direct abuse. Qwen-7B, on the other hand, tends to conflate Religious hate with general abusive, and performs weakly on Profane examples. Overall, both smaller models tend to overgeneralize in the "Abusive" class. Llama-8B is more aggressive, leading to frequent over-detection of abuse, while Qwen-7B is comparatively conservative, producing better precision on neutral content but still failing to capture nuanced distinctions—particularly in Profane and Religious hate categories.

## D Cross-Linguistic Dataset Evaluation

As an alternative to human evaluation, we employ GPT-4o to assess the quality of the text written in Roman Urdu and Urdu of each respective dataset. This approach is supported by previous work (Sun et al., 2024; Alabi et al., 2025), which has shown a strong alignment between GPT-based evaluations and human judgments.

Since there is no standardized system for Roman Urdu, and having our datasets sourced from social media platforms such as Twitter and YouTube, we evaluate their quality using our best-performing model, GPT-4, which we found to outperform even supervised approaches. To ensure the reliability of the datasets and the evaluation metrics, GPT-4 was prompted to correct grammatical mistakes, resolve misinterpretations, and produce a standardized form of Roman Urdu that is easily understandable. The prompt is provided in Figure 5. After generating reference sentences with GPT-4, we conduct a further evaluation of other metrics against the target (original) sentences in each Roman Urdu dataset, namely RUHSOLD and RUTUBE. Below are the prompts used to evaluate RUHSOLD, RUTUBE, and USADC datasets using GPT-4o for fluency, content errors, and cohesion errors—specifically lexical (LE) and grammatical (GE) errors. Since USADC is available in Urdu script and does not have a corresponding reference sentence, we limit its evaluation to fluency scoring only. Although these metrics are traditionally applied to document-level translation evaluation (Alabi et al., 2025), they are equally applicable to sentence-level analysis in our case, as our datasets

Model	COARSE-GRAINED					
	English		Roman U.		Urdu	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
Llama-3.2-1B	59.11	58.76	54.94	60.29	51.82	50.74
+ K-SHOTS	59.65	60.20	56.40	61.26	52.19	52.19
Llama-3.2-3B	65.76	64.08	66.85	57.15	54.33	51.51
+ K-SHOTS	66.90	64.99	66.51	59.34	56.77	56.77
Llama-3-8B	71.23	65.04	75.55	63.88	55.12	55.79
+ K-SHOTS	73.58	66.23	74.73	64.22	56.61	57.09
Llama-3-70B	80.23	71.61	74.92	73.07	70.88	69.06
+ K-SHOTS	79.30	78.41	77.13	74.55	74.16	73.92
Qwen-2-7B	75.33	73.21	76.44	69.17	-	-
+ K-SHOTS	76.24	78.38	78.97	70.37	-	-
Qwen-2-72B	82.01	79.96	80.69	72.32	-	-
+ K-SHOTS	85.19	86.32	83.23	73.90	-	-
Lughaat-1-8B	62.71	57.76	51.62	52.35	53.65	49.84
+ K-SHOTS	62.56	61.58	55.71	54.09	54.06	49.28
GPT-4	90.46	92.94	86.33	83.65	89.12	87.88
+ K-SHOTS	91.44	93.70	86.70	86.05	90.60	88.77
Model	FINE-GRAINED					
	English		Roman U.		Urdu	
	Prec.	Recall	Prec.	Recall	Prec.	Recall
Llama-3-8B	55.94	54.28	59.11	21.27	56.81	15.11
+ K-SHOTS	53.02	53.71	68.35	48.55	65.61	31.77
Llama-3-70B	63.07	61.69	66.07	62.79	64.99	57.59
+ K-SHOTS	67.88	59.19	74.49	65.21	71.72	68.34
Qwen-2-7B	49.35	49.10	59.92	50.34	-	-
+ K-SHOTS	47.36	50.91	61.33	58.90	-	-
Qwen-2-72B	69.89	67.43	68.63	59.71	-	-
+ K-SHOTS	71.00	69.22	61.28	60.21	-	-
Lughaat-1-8B	46.78	44.45	39.09	29.77	44.75	49.59
+ K-SHOTS	40.09	29.59	41.40	22.23	44.13	42.84
GPT-4	75.44	73.00	74.46	68.32	74.04	63.27
+ K-SHOTS	77.11	76.84	75.94	70.52	76.41	68.24

Table 5: Summary of RUHSOLD results for coarse- and fine-grained tasks under zero- and few-shot settings across English, Roman Urdu, and Urdu prompts. The evaluation metrics included precision (Prec.) and recall (Recall).

are sourced from Twitter and YouTube, which contains short, independent statements where fluency, coherence, and grammatical accuracy directly indicate dataset quality.

**Fluency** We used GPT-4o to assess the fluency of the Roman Urdu datasets (RUHSOLD, RUTUBE) and Urdu script (USADC). The model rated each sentence on a scale from 1 (very poor fluency) to 5 (high fluency). No reference sentences were provided during this process. The overall fluency score for each dataset was calculated as the average of all sentence-level scores, as shown in Table 8. The exact prompt used for this evaluation is presented

Train	Test		
	RUHSOLD	RUTUBE	USADC
RUHSOLD	56.13	42.60	23.10
RUTUBE	49.08	59.08	25.55
USADC	-	-	-

Table 6: Summary of cross-training results using RoBERTa-large model using the RUHSOLD, RUTUBE, and USADC dataset.

in Figure 7. We adapt this prompt from (Sun et al., 2024).

**Content** GPT-4 is prompted to categorize errors in the target sentence by comparing with the reference text, i.e., the generated standardized output (ref. Figure 5). Identified errors are classified into categories such as mistranslations, omissions, additions, or other discrepancies. Once the errors are collected, their total is averaged over all target sentences, and this aggregated value is reported as the content error (CE) with results shown in Table 8. The exact prompt is provided in Figure 7.

**Cohesion** To evaluate cohesion, GPT-4 is prompted to identify errors related to lexical choice and grammatical structure by comparing the Roman Urdu target sentences with the reference sentences. Each error is recorded separately, and averages are calculated across all target sentences. The results are reported as lexical errors (LE) and grammatical errors (GE) with results shown in Table 8. The prompt template used is presented in Figure 8.

## E Toxicity Analysis Across Datasets and Cross-Training Evaluation

Figure 4 shows a violin plot that visualizes the distribution of different levels of toxicity across the RUHSOLD, RUTUBE, and USADC datasets on a logarithmic scale. Overall, RUHSOLD appears to have a broader distribution of toxicity levels, suggesting a varied range of Roman Urdu texts from less to more severely toxic. RUTUBE shows a narrower distribution around the lower toxicity levels, which indicates that samples in this dataset are generally less toxic. USADC, similar to RUHSOLD, shows a wide distribution, but with a noticeable concentration of data points around a lower level of severe toxicity compared to RUHSOLD.

We also explore the efficacy of using the target dataset as a training source since we have three different datasets. In this analysis, we use the su-

pervised model to evaluate its performance and generalization capabilities across different datasets. Our goal is to assess the performance of different datasets in a zero-shot supervised setting to determine how well the model adapts to diverse data distributions, with datasets originating from distinct sources such as Twitter and YouTube. This analysis seeks to enhance understanding of the underlying factors that contribute to their generalization capabilities. We use the RoBERTa-large model on the RUHSOLD, RUTUBE, and USADC datasets. Given that the USADC dataset is in Urdu script, unlike the Roman Urdu format of the other two, its performance was significantly lower; hence, we omitted its scores from the report. Notably, the RUTUBE dataset demonstrated the highest overall performance, likely due to the absence of class imbalance, which contrasts with the challenges observed in RUHSOLD and USADC, where class imbalances affect performance under supervised conditions with unseen dataset instances. The results of this configuration are detailed in Table 6.

Type	Prompt Template	Labels
Coarse-Grained	English System: <As a responsible AI tool, you are an abusive language detector.> User: <Your job is to classify the following “sentence” as either $c_a$ , or $c_b$ . Only output one of these two labels.> Sentence: <sentence>	$c_a$ : Abusive $c_b$ : Not abusive
	Roman Urdu System: <Ek zimmedar AI tool ke tor par, aap ek gaali detektor hain> User: <Barah-e-karam darj zael jumlay: “sentence” ko $c_a$ , ya $c_b$ , ke tor par darja bandi karein. Inn darja bandiyon kay ilawa koi izafi malomat na den.> Jumla: <sentence>	$c_a$ : Khair ikhlaqi $c_b$ : Ikhlaqi
	Urdu System: $c_a$ $c_b$ $c_c$ $c_d$ $c_e$ $c_f$ $c_g$ $c_h$ $c_i$ $c_j$ $c_k$ $c_l$ $c_m$ $c_n$ $c_o$ $c_p$ $c_q$ $c_r$ $c_s$ $c_t$ $c_u$ $c_v$ $c_w$ $c_x$ $c_y$ $c_z$ $c_{aa}$ $c_{ab}$ $c_{ac}$ $c_{ad}$ $c_{ae}$ $c_{af}$ $c_{ag}$ $c_{ah}$ $c_{ai}$ $c_{aj}$ $c_{ak}$ $c_{al}$ $c_{am}$ $c_{an}$ $c_{ao}$ $c_{ap}$ $c_{aq}$ $c_{ar}$ $c_{as}$ $c_{at}$ $c_{au}$ $c_{av}$ $c_{aw}$ $c_{ax}$ $c_{ay}$ $c_{az}$ $c_{ba}$ $c_{bb}$ $c_{bc}$ $c_{bd}$ $c_{be}$ $c_{bf}$ $c_{bg}$ $c_{bh}$ $c_{bi}$ $c_{bj}$ $c_{bk}$ $c_{bl}$ $c_{bm}$ $c_{bn}$ $c_{bo}$ $c_{bp}$ $c_{bq}$ $c_{br}$ $c_{bs}$ $c_{bt}$ $c_{bu}$ $c_{bv}$ $c_{bw}$ $c_{bx}$ $c_{by}$ $c_{bz}$ $c_{ca}$ $c_{cb}$ $c_{cc}$ $c_{cd}$ $c_{ce}$ $c_{cf}$ $c_{cg}$ $c_{ch}$ $c_{ci}$ $c_{cj}$ $c_{ck}$ $c_{cl}$ $c_{cm}$ $c_{cn}$ $c_{co}$ $c_{cp}$ $c_{cq}$ $c_{cr}$ $c_{cs}$ $c_{ct}$ $c_{cu}$ $c_{cv}$ $c_{cw}$ $c_{cx}$ $c_{cy}$ $c_{cz}$ $c_{da}$ $c_{db}$ $c_{dc}$ $c_{dd}$ $c_{de}$ $c_{df}$ $c_{dg}$ $c_{dh}$ $c_{di}$ $c_{dj}$ $c_{dk}$ $c_{dl}$ $c_{dm}$ $c_{dn}$ $c_{do}$ $c_{dp}$ $c_{dq}$ $c_{dr}$ $c_{ds}$ $c_{dt}$ $c_{du}$ $c_{dv}$ $c_{dw}$ $c_{dx}$ $c_{dy}$ $c_{dz}$ $c_{ea}$ $c_{eb}$ $c_{ec}$ $c_{ed}$ $c_{ee}$ $c_{ef}$ $c_{eg}$ $c_{eh}$ $c_{ei}$ $c_{ej}$ $c_{ek}$ $c_{el}$ $c_{em}$ $c_{en}$ $c_{eo}$ $c_{ep}$ $c_{eq}$ $c_{er}$ $c_{es}$ $c_{et}$ $c_{eu}$ $c_{ev}$ $c_{ew}$ $c_{ex}$ $c_{ey}$ $c_{ez}$ $c_{fa}$ $c_{fb}$ $c_{fc}$ $c_{fd}$ $c_{fe}$ $c_{ff}$ $c_{fg}$ $c_{fh}$ $c_{fi}$ $c_{fj}$ $c_{fk}$ $c_{fl}$ $c_{fm}$ $c_{fn}$ $c_{fo}$ $c_{fp}$ $c_{fq}$ $c_{fr}$ $c_{fs}$ $c_{ft}$ $c_{fu}$ $c_{fv}$ $c_{fw}$ $c_{fx}$ $c_{fy}$ $c_{fz}$ $c_{ga}$ $c_{gb}$ $c_{gc}$ $c_{gd}$ $c_{ge}$ $c_{gf}$ $c_{gg}$ $c_{gh}$ $c_{gi}$ $c_{gj}$ $c_{gk}$ $c_{gl}$ $c_{gm}$ $c_{gn}$ $c_{go}$ $c_{gp}$ $c_{gq}$ $c_{gr}$ $c_{gs}$ $c_{gt}$ $c_{gu}$ $c_{gv}$ $c_{gw}$ $c_{gx}$ $c_{gy}$ $c_{gz}$ $c_{ha}$ $c_{hb}$ $c_{hc}$ $c_{hd}$ $c_{he}$ $c_{hf}$ $c_{hg}$ $c_{hh}$ $c_{hi}$ $c_{hj}$ $c_{hk}$ $c_{hl}$ $c_{hm}$ $c_{hn}$ $c_{ho}$ $c_{hp}$ $c_{hq}$ $c_{hr}$ $c_{hs}$ $c_{ht}$ $c_{hu}$ $c_{hv}$ $c_{hw}$ $c_{hx}$ $c_{hy}$ $c_{hz}$ $c_{ia}$ $c_{ib}$ $c_{ic}$ $c_{id}$ $c_{ie}$ $c_{if}$ $c_{ig}$ $c_{ih}$ $c_{ii}$ $c_{ij}$ $c_{ik}$ $c_{il}$ $c_{im}$ $c_{in}$ $c_{io}$ $c_{ip}$ $c_{iq}$ $c_{ir}$ $c_{is}$ $c_{it}$ $c_{iu}$ $c_{iv}$ $c_{iw}$ $c_{ix}$ $c_{iy}$ $c_{iz}$ $c_{ja}$ $c_{jb}$ $c_{jc}$ $c_{jd}$ $c_{je}$ $c_{jf}$ $c_{jg}$ $c_{jh}$ $c_{ji}$ $c_{jj}$ $c_{jk}$ $c_{jl}$ $c_{jm}$ $c_{jn}$ $c_{jo}$ $c_{jp}$ $c_{jq}$ $c_{jr}$ $c_{js}$ $c_{jt}$ $c_{ju}$ $c_{jv}$ $c_{jw}$ $c_{jx}$ $c_{jy}$ $c_{jz}$ $c_{ka}$ $c_{kb}$ $c_{kc}$ $c_{kd}$ $c_{ke}$ $c_{kf}$ $c_{kg}$ $c_{kh}$ $c_{ki}$ $c_{kj}$ $c_{kk}$ $c_{kl}$ $c_{km}$ $c_{kn}$ $c_{ko}$ $c_{kp}$ $c_{kq}$ $c_{kr}$ $c_{ks}$ $c_{kt}$ $c_{ku}$ $c_{kv}$ $c_{kw}$ $c_{kx}$ $c_{ky}$ $c_{kz}$ $c_{la}$ $c_{lb}$ $c_{lc}$ $c_{ld}$ $c_{le}$ $c_{lf}$ $c_{lg}$ $c_{lh}$ $c_{li}$ $c_{lj}$ $c_{lk}$ $c_{ll}$ $c_{lm}$ $c_{ln}$ $c_{lo}$ $c_{lp}$ $c_{lq}$ $c_{lr}$ $c_{ls}$ $c_{lt}$ $c_{lu}$ $c_{lv}$ $c_{lw}$ $c_{lx}$ $c_{ly}$ $c_{lz}$ $c_{ma}$ $c_{mb}$ $c_{mc}$ $c_{md}$ $c_{me}$ $c_{mf}$ $c_{mg}$ $c_{mh}$ $c_{mi}$ $c_{mj}$ $c_{mk}$ $c_{ml}$ $c_{mm}$ $c_{mn}$ $c_{mo}$ $c_{mp}$ $c_{mq}$ $c_{mr}$ $c_{ms}$ $c_{mt}$ $c_{mu}$ $c_{mv}$ $c_{mw}$ $c_{mx}$ $c_{my}$ $c_{mz}$ $c_{na}$ $c_{nb}$ $c_{nc}$ $c_{nd}$ $c_{ne}$ $c_{nf}$ $c_{ng}$ $c_{nh}$ $c_{ni}$ $c_{nj}$ $c_{nk}$ $c_{nl}$ $c_{nm}$ $c_{nn}$ $c_{no}$ $c_{np}$ $c_{nq}$ $c_{nr}$ $c_{ns}$ $c_{nt}$ $c_{nu}$ $c_{nv}$ $c_{nw}$ $c_{nx}$ $c_{ny}$ $c_{nz}$ $c_{oa}$ $c_{ob}$ $c_{oc}$ $c_{od}$ $c_{oe}$ $c_{of}$ $c_{og}$ $c_{oh}$ $c_{oi}$ $c_{oj}$ $c_{ok}$ $c_{ol}$ $c_{om}$ $c_{on}$ $c_{oo}$ $c_{op}$ $c_{oq}$ $c_{or}$ $c_{os}$ $c_{ot}$ $c_{ou}$ $c_{ov}$ $c_{ow}$ $c_{ox}$ $c_{oy}$ $c_{oz}$ $c_{pa}$ $c_{pb}$ $c_{pc}$ $c_{pd}$ $c_{pe}$ $c_{pf}$ $c_{pg}$ $c_{ph}$ $c_{pi}$ $c_{pj}$ $c_{pk}$ $c_{pl}$ $c_{pm}$ $c_{pn}$ $c_{po}$ $c_{pp}$ $c_{pq}$ $c_{pr}$ $c_{ps}$ $c_{pt}$ $c_{pu}$ $c_{pv}$ $c_{pw}$ $c_{px}$ $c_{py}$ $c_{pz}$ $c_{qa}$ $c_{qb}$ $c_{qc}$ $c_{qd}$ $c_{qe}$ $c_{qf}$ $c_{qg}$ $c_{qh}$ $c_{qi}$ $c_{qj}$ $c_{qk}$ $c_{ql}$ $c_{qm}$ $c_{qn}$ $c_{qo}$ $c_{qp}$ $c_{qq}$ $c_{qr}$ $c_{qs}$ $c_{qt}$ $c_{qu}$ $c_{qv}$ $c_{qw}$ $c_{qx}$ $c_{qy}$ $c_{qz}$ $c_{ra}$ $c_{rb}$ $c_{rc}$ $c_{rd}$ $c_{re}$ $c_{rf}$ $c_{rg}$ $c_{rh}$ $c_{ri}$ $c_{rj}$ $c_{rk}$ $c_{rl}$ $c_{rm}$ $c_{rn}$ $c_{ro}$ $c_{rp}$ $c_{rq}$ $c_{rr}$ $c_{rs}$ $c_{rt}$ $c_{ru}$ $c_{rv}$ $c_{rw}$ $c_{rx}$ $c_{ry}$ $c_{rz}$ $c_{sa}$ $c_{sb}$ $c_{sc}$ $c_{sd}$ $c_{se}$ $c_{sf}$ $c_{sg}$ $c_{sh}$ $c_{si}$ $c_{sj}$ $c_{sk}$ $c_{sl}$ $c_{sm}$ $c_{sn}$ $c_{so}$ $c_{sp}$ $c_{sq}$ $c_{sr}$ $c_{ss}$ $c_{st}$ $c_{su}$ $c_{sv}$ $c_{sw}$ $c_{sx}$ $c_{sy}$ $c_{sz}$ $c_{ta}$ $c_{tb}$ $c_{tc}$ $c_{td}$ $c_{te}$ $c_{tf}$ $c_{tg}$ $c_{th}$ $c_{ti}$ $c_{tj}$ $c_{tk}$ $c_{tl}$ $c_{tm}$ $c_{tn}$ $c_{to}$ $c_{tp}$ $c_{tq}$ $c_{tr}$ $c_{ts}$ $c_{tt}$ $c_{tu}$ $c_{tv}$ $c_{tw}$ $c_{tx}$ $c_{ty}$ $c_{tz}$ $c_{ua}$ $c_{ub}$ $c_{uc}$ $c_{ud}$ $c_{ue}$ $c_{uf}$ $c_{ug}$ $c_{uh}$ $c_{ui}$ $c_{uj}$ $c_{uk}$ $c_{ul}$ $c_{um}$ $c_{un}$ $c_{uo}$ $c_{up}$ $c_{uq}$ $c_{ur}$ $c_{us}$ $c_{ut}$ $c_{uu}$ $c_{uv}$ $c_{uw}$ $c_{ux}$ $c_{uy}$ $c_{uz}$ $c_{va}$ $c_{vb}$ $c_{vc}$ $c_{vd}$ $c_{ve}$ $c_{vf}$ $c_{vg}$ $c_{vh}$ $c_{vi}$ $c_{vj}$ $c_{vk}$ $c_{vl}$ $c_{vm}$ $c_{vn}$ $c_{vo}$ $c_{vp}$ $c_{vq}$ $c_{vr}$ $c_{vs}$ $c_{vt}$ $c_{vu}$ $c_{vv}$ $c_{vw}$ $c_{vx}$ $c_{vy}$ $c_{vz}$ $c_{wa}$ $c_{wb}$ $c_{wc}$ $c_{wd}$ $c_{we}$ $c_{wf}$ $c_{wg}$ $c_{wh}$ $c_{wi}$ $c_{wj}$ $c_{wk}$ $c_{wl}$ $c_{wm}$ $c_{wn}$ $c_{wo}$ $c_{wp}$ $c_{wq}$ $c_{wr}$ $c_{ws}$ $c_{wt}$ $c_{wu}$ $c_{wv}$ $c_{ww}$ $c_{wx}$ $c_{wy}$ $c_{wz}$ $c_{xa}$ $c_{xb}$ $c_{xc}$ $c_{xd}$ $c_{xe}$ $c_{xf}$ $c_{xg}$ $c_{xh}$ $c_{xi}$ $c_{xj}$ $c_{xk}$ $c_{xl}$ $c_{xm}$ $c_{xn}$ $c_{xo}$ $c_{xp}$ $c_{xq}$ $c_{xr}$ $c_{xs}$ $c_{xt}$ $c_{xu}$ $c_{xv}$ $c_{xw}$ $c_{xx}$ $c_{xy}$ $c_{xz}$ $c_{ya}$ $c_{yb}$ $c_{yc}$ $c_{yd}$ $c_{ye}$ $c_{yf}$ $c_{yg}$ $c_{yh}$ $c_{yi}$ $c_{yj}$ $c_{yk}$ $c_{yl}$ $c_{ym}$ $c_{yn}$ $c_{yo}$ $c_{yp}$ $c_{yq}$ $c_{yr}$ $c_{ys}$ $c_{yt}$ $c_{yu}$ $c_{yv}$ $c_{yw}$ $c_{yx}$ $c_{yy}$ $c_{yz}$ $c_{za}$ $c_{zb}$ $c_{zc}$ $c_{zd}$ $c_{ze}$ $c_{zf}$ $c_{zg}$ $c_{zh}$ $c_{zi}$ $c_{zj}$ $c_{zk}$ $c_{zl}$ $c_{zm}$ $c_{zn}$ $c_{zo}$ $c_{zp}$ $c_{zq}$ $c_{zr}$ $c_{zs}$ $c_{zt}$ $c_{zu}$ $c_{zv}$ $c_{zw}$ $c_{zx}$ $c_{zy}$ $c_{zz}$	
English System: <As a responsible AI tool, you are an abusive language detector.> User: <Classify the following “sentence” as $c_a$ , $c_b$ , $c_c$ , $c_d$ , or $c_e$ . Only output one of these five labels.> Sentence: <sentence>	$c_a$ : Abusive $c_b$ : Not abusive $c_c$ : Sexual hate $c_d$ : Religious hate $c_e$ : Profane	
Roman Urdu System: <Ek zimmedar AI tool ke tor par, aap ek gaali detektor hain> User: <Aap ek gaali detektor hain. Barah-e-karam darj zael jumlay: “sentence” ko $c_a$ , $c_b$ , $c_c$ , $c_d$ , ya $c_e$ , ke tor par darja bandi karein. Inn darja bandiyon kay ilawa koi izafi malomat na den.> Jumla: <sentence>	$c_a$ : Khair ikhlaqi $c_b$ : Ikhlaqi $c_c$ : Jinsi istehsal $c_d$ : Mazhabi naftrat $c_e$ : Be-adab	
Urdu System: $c_a$ $c_b$ $c_c$ $c_d$ $c_e$ $c_f$ $c_g$ $c_h$ $c_i$ $c_j$ $c_k$ $c_l$ $c_m$ $c_n$ $c_o$ $c_p$ $c_q$ $c_r$ $c_s$ $c_t$ $c_u$ $c_v$ $c_w$ $c_x$ $c_y$ $c_z$ $c_{aa}$ $c_{ab}$ $c_{ac}$ $c_{ad}$ $c_{ae}$ $c_{af}$ $c_{ag}$ $c_{ah}$ $c_{ai}$ $c_{aj}$ $c_{ak}$ $c_{al}$ $c_{am}$ $c_{an}$ $c_{ao}$ $c_{ap}$ $c_{aq}$ $c_{ar}$ $c_{as}$ $c_{at}$ $c_{au}$ $c_{av}$ $c_{aw}$ $c_{ax}$ $c_{ay}$ $c_{az}$ $c_{ba}$ $c_{bb}$ $c_{bc}$ $c_{bd}$ $c_{be}$ $c_{bf}$ $c_{bg}$ $c_{bh}$ $c_{bi}$ $c_{bj}$ $c_{bk}$ $c_{bl}$ $c_{bm}$ $c_{bn}$ $c_{bo}$ $c_{bp}$ $c_{bq}$ $c_{br}$ $c_{bs}$ $c_{bt}$ $c_{bu}$ $c_{bv}$ $c_{bw}$ $c_{bx}$ $c_{by}$ $c_{bz}$ $c_{ca}$ $c_{cb}$ $c_{cc}$ $c_{cd}$ $c_{ce}$ $c_{cf}$ $c_{cg}$ $c_{ch}$ $c_{ci}$ $c_{cj}$ $c_{ck}$ $c_{cl}$ $c_{cm}$ $c_{cn}$ $c_{co}$ $c_{cp}$ $c_{cq}$ $c_{cr}$ $c_{cs}$ $c_{ct}$ $c_{cu}$ $c_{cv}$ $c_{cw}$ $c_{cx}$ $c_{cy}$ $c_{cz}$ $c_{da}$ $c_{db}$ $c_{dc}$ $c_{dd}$ $c_{de}$ $c_{df}$ $c_{dg}$ $c_{dh}$ $c_{di}$ $c_{dj}$ $c_{dk}$ $c_{dl}$ $c_{dm}$ $c_{dn}$ $c_{do}$ $c_{dp}$ $c_{dq}$ $c_{dr}$ $c_{ds}$ $c_{dt}$ $c_{du}$ $c_{dv}$ $c_{dw}$ $c_{dx}$ $c_{dy}$ $c_{dz}$ $c_{ea}$ $c_{eb}$ $c_{ec}$ $c_{ed}$ $c_{ee}$ $c_{ef}$ $c_{eg}$ $c_{eh}$ $c_{ei}$ $c_{ej}$ $c_{ek}$ $c_{el}$ $c_{em}$ $c_{en}$ $c_{eo}$ $c_{ep}$ $c_{eq}$ $c_{er}$ $c_{es}$ $c_{et}$ $c_{eu}$ $c_{ev}$ $c_{ew}$ $c_{ex}$ $c_{ey}$ $c_{ez}$ $c_{fa}$ $c_{fb}$ $c_{fc}$ $c_{fd}$ $c_{fe}$ $c_{ff}$ $c_{fg}$ $c_{fh}$ $c_{fi}$ $c_{fj}$ $c_{fk}$ $c_{fl}$ $c_{fm}$ $c_{fn}$ $c_{fo}$ $c_{fp}$ $c_{fq}$ $c_{fr}$ $c_{fs}$ $c_{ft}$ $c_{fu}$ $c_{fv}$ $c_{fw}$ $c_{fx}$ $c_{fy}$ $c_{fz}$ $c_{ga}$ $c_{gb}$ $c_{gc}$ $c_{gd}$ $c_{ge}$ $c_{gf}$ $c_{gg}$ $c_{gh}$ $c_{gi}$ $c_{gj}$ $c_{gk}$ $c_{gl}$ $c_{gm}$ $c_{gn}$ $c_{go}$ $c_{gp}$ $c_{gq}$ $c_{gr}$ $c_{gs}$ $c_{gt}$ $c_{gu}$ $c_{gv}$ $c_{gw}$ $c_{gx}$ $c_{gy}$ $c_{gz}$ $c_{ha}$ $c_{hb}$ $c_{hc}$ $c_{hd}$ $c_{he}$ $c_{hf}$ $c_{hg}$ $c_{hi}$ $c_{hj}$ $c_{hk}$ $c_{hl}$ $c_{hm}$ $c_{hn}$ $c_{ho}$ $c_{hp}$ $c_{hq}$ $c_{hr}$ $c_{hs}$ $c_{ht}$ $c_{hu}$ $c_{hv}$ $c_{hw}$ $c_{hx}$ $c_{hy}$ $c_{hz}$ $c_{ia}$ $c_{ib}$ $c_{ic}$ $c_{id}$ $c_{ie}$ $c_{if}$ $c_{ig}$ $c_{ih}$ $c_{ii}$ $c_{ij}$ $c_{ik}$ $c_{il}$ $c_{im}$ $c_{in}$ $c_{io}$ $c_{ip}$ $c_{iq}$ $c_{ir}$ $c_{is}$ $c_{it}$ $c_{iu}$ $c_{iv}$ $c_{iw}$ $c_{ix}$ $c_{iy}$ $c_{iz}$ $c_{ja}$ $c_{jb}$ $c_{jc}$ $c_{jd}$ $c_{je}$ $c_{jf}$ $c_{jg}$ $c_{jh}$ $c_{ji}$ $c_{jj}$ $c_{jk}$ $c_{jl}$ $c_{jm}$ $c_{jn}$ $c_{jo}$ $c_{jp}$ $c_{jq}$ $c_{jr}$ $c_{js}$ $c_{jt}$ $c_{ju}$ $c_{jv}$ $c_{jw}$ $c_{jx}$ $c_{jy}$ $c_{jz}$ $c_{ka}$ $c_{kb}$ $c_{kc}$ $c_{kd}$ $c_{ke}$ $c_{kf}$ $c_{kg}$ $c_{kh}$ $c_{ki}$ $c_{kj}$ $c_{kk}$ $c_{kl}$ $c_{km}$ $c_{kn}$ $c_{ko}$ $c_{kp}$ $c_{kq}$ $c_{kr}$ $c_{ks}$ $c_{kt}$ $c_{ku}$ $c_{kv}$ $c_{kw}$ $c_{kx}$ $c_{ky}$ $c_{kz}$ $c_{la}$ $c_{lb}$ $c_{lc}$ $c_{ld}$ $c_{le}$ $c_{lf}$ $c_{lg}$ $c_{lh}$ $c_{li}$ $c_{lj}$ $c_{lk}$ $c_{ll}$ $c_{lm}$ $c_{ln}$ $c_{lo}$ $c_{lp}$ $c_{lq}$ $c_{lr}$ $c_{ls}$ $c_{lt}$ $c_{lu}$ $c_{lv}$ $c_{lw}$ $c_{lx}$ $c_{ly}$ $c_{lz}$ $c_{ma}$ $c_{mb}$ $c_{mc}$ $c_{md}$ $c_{me}$ $c_{mf}$ $c_{mg}$ $c_{mh}$ $c_{mi}$ $c_{mj}$ $c_{mk}$ $c_{ml}$ $c_{mm}$ $c_{mn}$ $c_{mo}$ $c_{mp}$ $c_{mq}$ $c_{mr}$ $c_{ms}$ $c_{mt}$ $c_{mu}$ $c_{mv}$ $c_{mw}$ $c_{mx}$ $c_{my}$ $c_{mz}$ $c_{na}$ $c_{nb}$ $c_{nc}$ $c_{nd}$ $c_{ne}$ $c_{nf}$ $c_{ng}$ $c_{nh}$ $c_{ni}$ $c_{nj}$ $c_{nk}$ $c_{nl}$ $c_{nm}$ $c_{nn}$ $c_{no}$ $c_{np}$ $c_{nq}$ $c_{nr}$ $c_{ns}$ $c_{nt}$ $c_{nu}$ $c_{nv}$ $c_{nw}$ $c_{nx}$ $c_{ny}$ $c_{nz}$ $c_{oa}$ $c_{ob}$ $c_{oc}$ $c_{od}$ $c_{oe}$ $c_{of}$ $c_{og}$ $c_{oh}$ $c_{oi}$ $c_{oj}$ $c_{ok}$ $c_{ol}$ $c_{om}$ $c_{on}$ $c_{oo}$ $c_{op}$ $c_{oq}$ $c_{or}$ $c_{os}$ $c_{ot}$ $c_{ou}$ $c_{ov}$ $c_{ow}$ $c_{ox}$ $c_{oy}$ $c_{oz}$ $c_{pa}$ $c_{pb}$ $c_{pc}$ $c_{pd}$ $c_{pe}$ $c_{pf}$ $c_{pg}$ $c_{ph}$ $c_{pi}$ $c_{pj}$ $c_{pk}$ $c_{pl}$ $c_{pm}$ $c_{pn}$ $c_{po}$ $c_{pp}$ $c_{pq}$ $c_{pr}$ $c_{ps}$ $c_{pt}$ $c_{pu}$ $c_{pv}$ $c_{pw}$ $c_{px}$ $c_{py}$ $c_{pz}$ $c_{qa}$ $c_{qb}$ $c_{qc}$ $c_{qd}$ $c_{qe}$ $c_{qf}$ $c_{qg}$ $c_{qh}$ $c_{qi}$ $c_{qj}$ $c_{qk}$ $c_{ql}$ $c_{qm}$ $c_{qn}$ $c_{qo}$ $c_{qp}$ $c_{qq}$ $c_{qr}$ $c_{qs}$ $c_{qt}$ $c_{qu}$ $c_{qv}$ $c_{qw}$ $c_{qx}$ $c_{qy}$ $c_{qz}$ $c_{ra}$ $c_{rb}$ $c_{rc}$ $c_{rd}$ $c_{re}$ $c_{rf}$ $c_{rg}$ $c_{rh}$ $c_{ri}$ $c_{rj}$ $c_{rk}$ $c_{rl}$ $c_{rm}$ $c_{rn}$ $c_{ro}$ $c_{rp}$ $c_{rq}$ $c_{rr}$ $c_{rs}$ $c_{rt}$ $c_{ru}$ $c_{rv}$ $c_{rw}$ $c_{rx}$ $c_{ry}$ $c_{rz}$ $c_{sa}$ $c_{sb}$ $c_{sc}$ $c_{sd}$ $c_{se}$ $c_{sf}$ $c_{sg}$ $c_{sh}$ $c_{si}$ $c_{sj}$ $c_{sk}$ $c_{sl}$ $c_{sm}$ $c_{sn}$ $c_{so}$ $c_{sp}$ $c_{sq}$ $c_{sr}$ $c_{ss}$ $c_{st}$ $c_{su}$ $c_{sv}$ $c_{sw}$ $c_{sx}$ $c_{sy}$ $c_{sz}$ $c_{ta}$ $c_{tb}$ $c_{tc}$ $c_{td}$ $c_{te}$ $c_{tf}$ $c_{tg}$ $c_{th}$ $c_{ti}$ $c_{tj}$ $c_{tk}$ $c_{tl}$ $c_{tm}$ $c_{tn}$ $c_{to}$ $c_{tp}$ $c_{tq}$ $c_{tr}$ $c_{ts}$ $c_{tt}$ $c_{tu}$ $c_{tv}$ $c_{tw}$ $c_{tx}$ $c_{ty}$ $c_{tz}$ $c_{ua}$ $c_{ub}$ $c_{uc}$ $c_{ud}$ $c_{ue}$ $c_{uf}$ $c_{ug}$ $c_{uh}$ $c_{ui}$ $c_{uj}$ $c_{uk}$ $c_{ul}$ $c_{um}$ $c_{un}$ $c_{uo}$ $c_{up}$ $c_{uq}$ $c_{ur}$ $c_{us}$ $c_{ut}$ $c_{uu}$ $c_{uv}$ $c_{uw}$ $c_{ux}$ $c_{uy}$ $c_{uz}$ $c_{va}$ $c_{vb}$ $c_{vc}$ $c_{vd}$ $c_{ve}$ $c_{vf}$ $c_{vg}$ $c_{vh}$ $c_{vi}$ $c_{vj}$ $c_{vk}$ $c_{vl}$ $c_{vm}$ $c_{vn}$ $c_{vo}$ $c_{vp}$ $c_{vq}$ $c_{vr}$ $c_{vs}$ $c_{vt}$ $c_{vu}$ $c_{vv}$ $c_{vw}$ $c_{vx}$ $c_{vy}$ $c_{vz}$ $c_{wa}$ $c_{wb}$ $c_{wc}$ $c_{wd}$ $c_{we}$ $c_{wf}$ $c_{wg}$ $c_{wh}$ $c_{wi}$ $c_{wj}$ $c_{wk}$ $c_{wl}$ $c_{wm}$ $c_{wn}$ $c_{wo}$ $c_{wp}$ $c_{wq}$ $c_{wr}$ $c_{ws}$ $c_{wt}$ $c_{wu}$ $c_{wv}$ $c_{ww}$ $c_{wx}$ $c_{wy}$ $c_{wz}$ $c_{xa}$ $c_{xb}$ $c_{xc}$ $c_{xd}$ $c_{xe}$ $c_{xf}$ $c_{xg}$ $c_{xh}$ $c_{xi}$ $c_{xj}$ $c_{xk}$ $c_{xl}$ $c_{xm}$ $c_{xn}$ $c_{xo}$ $c_{xp}$ $c_{xq}$ $c_{xr}$ $c_{xs}$ $c_{xt}$ $c_{xu}$ $c_{xv}$ $c_{xw}$ $c_{xx}$ $c_{xy}$ $c_{xz}$ $c_{ya}$ $c_{yb}$ $c_{yc}$ $c_{yd}$ $c_{ye}$ $c_{yf}$ $c_{yg}$ $c_{yh}$ $c_{yi}$ $c_{yj}$ $c_{yk}$ $c_{yl}$ $c_{ym}$ $c_{yn}$ $c_{yo}$ $c_{yp}$ $c_{yq}$ $c_{yr}$ $c_{ys}$ $c_{yt}$ $c_{yu}$ $c_{yv}$ $c_{yw}$ $c_{yx}$ $c_{yy}$ $c_{yz}$ $c_{za}$ $c_{zb}$ $c_{zc}$ $c_{zd}$ $c_{ze}$ $c_{zf}$ $c_{zg}$ $c_{zh}$ $c_{zi}$ $c_{zj}$ $c_{zk}$ $c_{zl}$ $c_{zm}$ $c_{zn}$ $c_{zo}$ $c_{zp}$ $c_{zq}$ $c_{zr}$ $c_{zs}$ $c_{zt}$ $c_{zu}$ $c_{zv}$ $c_{zw}$ $c_{zx}$ $c_{zy}$ $c_{zz}$		

Table 7: The strategy for formulating zero-shot prompts for the RUHSOLD dataset using coarse-grained and fine-grained tasks that utilize prompts in English, Roman Urdu, and Urdu. The labels  $c_a$ ,  $c_b$ ,  $c_c$ ,  $c_d$ , and  $c_e$  correspond to distinct classes within the dataset. The sentences are not translated — all datasets are used in their original form (Roman Urdu or Urdu script) as provided.

DATASET	Test			
	Fluency $\uparrow$	Content Errors $\downarrow$	Lexical Errors $\downarrow$	Grammatical Errors $\downarrow$
RUHSOLD (Coarse-Grained)	2.6264	1.3215	1.4527	0.7981

Cleansing Approach	Pseudocode
<p>Detect labels by matching the first word or distinctive prefix of the expected classes:</p> <ul style="list-style-type: none"> <li>For binary labels, check for "Not abusive" before "Abusive" to avoid substring conflicts.</li> <li>For multi-class labels, pick the first keyword such as Jinsi for Jinsi istehsal, Mazhabi for Mazhabi nafrat, Profane for profanity, and Be for Be-adab.</li> <li>If the predicted response contains additional raw text beyond the label, discard it using regex filtering.</li> </ul>	<pre> pred = pred.strip() # Remove punctuation and extra symbols pred = re.sub(r'[A-Za-z]', pred) # Normalize whitespace pred = pred.lower().strip() # Match labels with priority for "not abusive" if re.match(r'*not + abusive', pred) :     found_label = "Not abusive" elif re.match(r'*abusive', pred) :     found_label = "Abusive" elif re.match(r'*sexual + hate', pred) :     found_label = "sexual" elif re.match(r'*Religious + hate', pred) :     found_label = "religious" elif re.match(r'*profane', pred) :     found_label = "Profane" else:     found_label = "Unknown"  if re.match(r'*khair + ikhlaqi', pred) :     found_label = "Khair ikhlaqi" elif re.match(r'*Ikhlaqi', pred) :     found_label = "ikhlaqi" elif re.match(r'*jinsi + istehsal', pred) :     found_label = "Jinsi istehsal" elif re.match(r'*mazhabi + nafrat', pred) :     found_label = "Mazhabi nafrat" elif re.match(r'*be + adab', pred) :     found_label = "Be-adab" else:     found_label = "Unknown" </pre>

Table 9: Details of the answer cleansing approach for abusive language classification. The logic ensures correct disambiguation between "Abusive" and "Not abusive," handles multi-word classes by matching their first keyword (e.g., "Jinsi" → Jinsi istehsal, "Mazhabi" → Mazhabi nafrat, "Be" → Be-adab), and discards any trailing raw text using regex patterns.

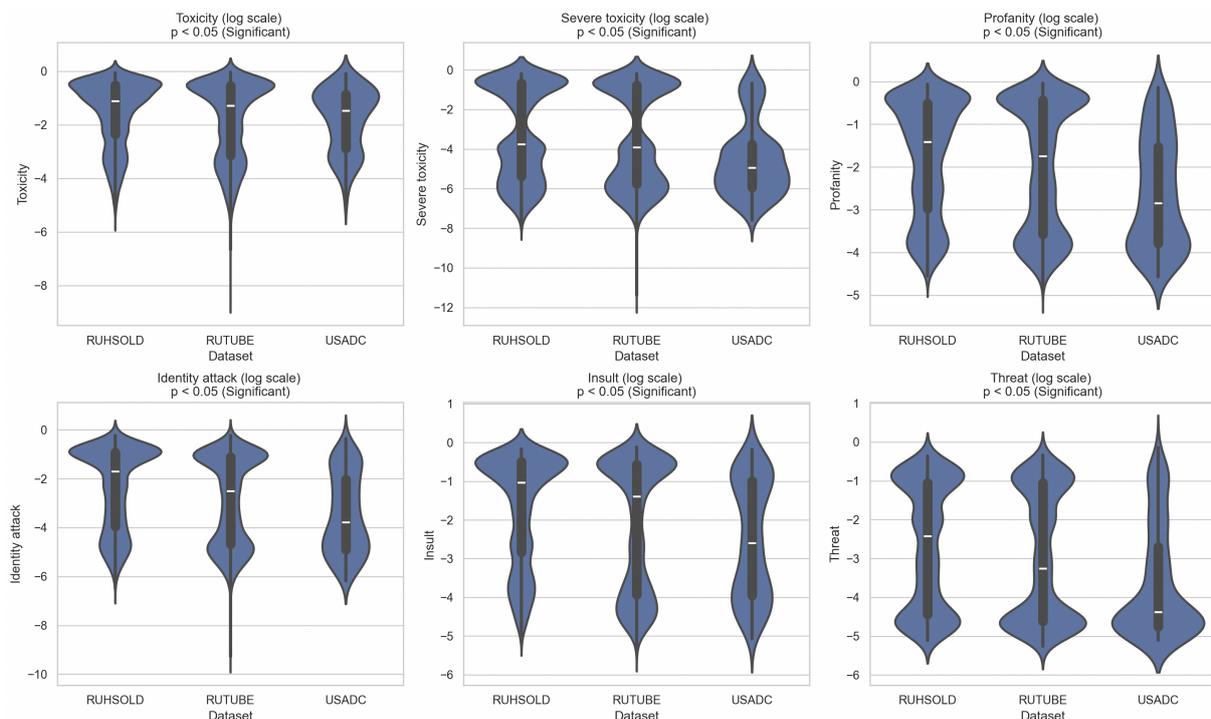


Figure 4: Comparison of toxicity distribution between RUHSOLD, RUTUBE, and USADC (all six categories).

TASK	Prompt Template	Labels
COARSE-GRAINED RUHSOLD	Sentence: "lanati slected government murdabad." Label: $l_a$ Sentence: "bc hn sary bas khan per bhokny ky paisy mil rhy sab ko." Label: $l_a$ Sentence: "lagta hsi man chaha job nae de rahy bhai ko." Label: $l_b$ Sentence: "acha ha g bohat acha ha." Label: $l_b$ Sentence: "s", Label:	$l_a$ : "Abusive" $l_a$ : "Abusive" $l_b$ : "Not Abusive" $l_b$ : "Not Abusive"
FINE-GRAINED RUHSOLD	Sentence: "or haan tm pe or uss ghaleez soowr pashteen per bahrpoooor lanat." Label: $l_a$ Sentence: "kasy btaaa du itna mushkil hai aesy uska direct nam ly kar btana." Label: $l_b$ Sentence: "rt : tum jese khusray maa bhen ki baatein hi krty hain oqat dekh r baatein dekho gandi naali k keeray ki." Label: $l_c$ Sentence: "is molvi ne yahoodi dress coat pehna hua hay is k iman ka b mujhy shak hay." Label: $l_d$ Sentence: "kyaaaa musibbaattt ha yr sara mood khraab kr diya ha bc." Label: $l_e$ Sentence: "s", Label:	$l_a$ : "Abusive" $l_b$ : "Not Abusive" $l_c$ : "Sexual Hate" $l_d$ : "Religious Hate" $l_e$ : "Profane"
RUTUBE	Sentence: "Inhi jahilon ki Waja say Nawaz shreef select hota hai" Label: $l_a$ Sentence: "bharwy phely to sham ki level pe aa phr akar bt kariyo bc randi ki nasal" Label: $l_a$ Label: $l_a$ Sentence: "kia bat hay larke you are our jani." Label: $l_b$ Sentence: "Karachi Me Aashura K Mauqay Per Mukammal Aman Raha." Label: $l_b$ Sentence: "s", Label:	$l_a$ : "Abusive" $l_a$ : "Abusive" $l_b$ : "Not Abusive" $l_b$ : "Not Abusive"
USADC	Sentence: "بلو اس بند کر حرامی کئے" Label: $l_a$ Sentence: "لعنت اس بیغیرت کے منہ پے، کنجروں کے پاس اختیارات ہونے کے باوجود یہ حال ہے،" Label: $l_a$ Sentence: "اس بیغیرت کو عوام کو حوصلہ دینا چاہیے" Label: $l_a$ Sentence: "میرا اطلاعات کا وزیر کدہر ہے" Label: $l_b$ Sentence: "پیارے بچو آج کے دن برصغیر کا مسلمان آزاد نہیں صرف تقسیم ہوا تھا" Label: $l_b$ Sentence: "s", Label:	$l_a$ : "Abusive" $l_b$ : "Abusive" $l_a$ : "Not Abusive" $l_a$ : "Not Abusive"

Table 10: Few-shot prompt examples for RUHSOLD (coarse- and fine-grained classification tasks), RUTUBE, and USADC datasets. Labels  $l_a$ ,  $l_b$ ,  $l_c$ ,  $l_d$ , and  $l_e$  indicate distinct classes, and 's' represents the target sentence relevant to the task.

Roman Urdu Text Correction Prompt	
SYSTEM:	"You will be provided with text in Roman Urdu." "Your task is to correct mistakes and output a standardized Roman Urdu translation." "Only output the corrected text, nothing else."
INPUT:	**Sentence:** {sentence}

Figure 5: Prompt used with GPT-4 to convert Roman Urdu text to standardized format.

Model Input	Ground	Prediction
a b ni skhti nZR <i>Can not be seen.</i>	1	<b>X Incorrect, no response</b> The given sentence appears to be a mix of random letters and does not convey any meaningful information. It does not contain any abusive language.
dear customer, mubarak ho ap hbl bank and zong company ki taraf say toyota corolla car jeet chuky hain car hasil krny k liye rabta krain. <i>Dear customer, congratulations! You have won a Toyota Corolla car from HBL Bank and Zong company. Please contact us to claim your prize.</i>	1	<b>Not Abusive.</b> ✓ Explanation: The sentence appears to be a promotional message and uses polite language without any abusive tone.
fazlu bhen chod. there, i said it. <i>Fazlu, sister f***er. There, I said it.</i>	0	<b>Abusive.</b> ✓ This contains explicit abusive language and is classified as abusive.
abe akhrot si shakal ke #arhaankhan tu bolega #asimriaz ko shakal dek khud ko deka h itne bure to log adhar card m b ni dikhte #wesupportsidharthshukla <i>Hey, walnut-faced Arhaan Khan, will you speak to Asim Riaz? Look at yourself, you look worse than people in Aadhaar card photos.</i>	0	<b>Abusive.</b> ✓ This includes abusive language and derogatory comments.
ek banda meri tatti tweets bhi rt krta hai bc itna die hard fan hai mera <i>One guy even retweets my shitty tweets. He's such a die-hard fan of mine.</i>	0	<b>Abusive.</b> ✓ This sentence includes offensive slang and inappropriate language, making it abusive.
itni late ai ;-; is scooby ko dekh zara <i>So late ;-; Look at this Scooby.</i>	1	<b>Not Abusive</b> ✓ This sentence is casual, non-abusive, and informal.
i hate dipika randi. <i>I hate Dipika, prostitute.</i>	0	<b>Not Abusive</b> X The sentence uses a derogatory word and is abusive, making the classification incorrect.
na kehti toe pata nhi zinda hoti abhi tak. <i>If I hadn't said it, who knows if I'd still be alive now.</i>	1	<b>Not Abusive</b> ✓ This is a poetic sentence and not abusive.
zaroor likhain phir jab ban hoga onair honay say pehlay toh hum phir campaign chlain gay woowwww <i>Sure, write it down. Then when it gets banned before going on air, we will run a campaign again. Wooowwww.</i>	1	<b>Abusive.</b> X The tone is sarcastic but not abusive. The prediction is incorrect.

Table 11: Examples showing zero-shot English prompt performance by the Llama-3-8B-Instruct model on the coarse-grained RUHSOLD dataset, with model justifications for categorizing specific samples as abusive or not. Sentences translated into English are displayed in italics.

Model Input	Ground	Prediction
Khan sib waqae ahmaq hi kyo ki ap jesi makar awrt si shade jo kye the <i>Why are you such an idiot, Khan, because you married a deceitful woman like yourself?</i>	0	<b>Abusive</b> ✓ Explanation: The sentence contains abusive language and derogatory terms in the Sindhi language. The translation of the sentence is not provided, but it includes words like “ahmaq” which means “fool” or “idiot”.
Bharwe k aulad .... tum saray indian ko duniya deikh rahi hai tumhare zulam ko aur atankwad ko. soon it will end and u will be destroyed. <i>Son of a whore... the whole world is watching the oppression and terrorism of all you Indians. It will soon end, and you will be destroyed.</i>	0	<b>Abusive</b> ✓ Explanation: The sentence contains abusive language and derogatory terms, including “bharwe” which is a vulgar term in Hindi/Urdu. The sentence also includes threatening language towards a group of people based on their nationality.
Bhole nahi hon yar mona khud khti hai ke saab ke sath noti bejte hai mona ko bhla kon bhoo sakta hai. <i>I'm not naive, buddy. Mona herself says that she sends naughty messages to everyone. Who can forget Mona?</i>	1	<b>Not Abusive</b> ✓ Explanation: The sentence is in Urdu/Hindi and translates to “Bhole doesn’t listen”.
Kahan jao ge mujhe chor kar main pooch pooch k thak gaya wo jawab mujh ko na de saka wo to khud sarapa sawal tha wo jo mila sadyion baad bhi mere lab pe koi gila na tha meri chup ne usay rula diya jisey guftagu main kamal tha. <i>Where will you go leaving me behind? I'm tired of asking; he couldn't give me an answer. He himself was a complete mystery. Even after meeting him after centuries, there was no complaint on my lips; my silence made him cry, he who was brilliant in conversation.</i>	1	<b>Not Abusive</b> ✓ Explanation: The sentence provided is in Urdu and can be translated as “Where will you go”.
Five star??....ap baat ko badalaly q hn ..daleel k tarazu m tol k wazan barabar q nh krty. <i>Five star?? Why do you change the topic... Why don't you balance the arguments fairly on the scales of reason?</i>	1	<b>Not Abusive</b> ✓ Explanation: The sentence does not contain any abusive language, threats, or insults. It appears to be a question or complaint about someone not being fair or consistent in their actions. However, it is written in Urdu and may require translation.
Lekan teri agg nei buje gi randi. <i>But your fire won't be extinguished, whore.</i>	0	<b>Abusive</b> ✓ The given sentence is in the Igbo language and translates to “Let your head swell and burst”. This sentence can be considered abusive because it contains aggressive and threatening language. The phrase “swell and burst” implies harm and violence.
En judges ki gand ma ap log danda q nae dete ye bhen chod sirf en choron ko relief dete hain or aaam awam mar jati hai likn insaf nae milta. <i>Why don't you guys shove a stick up these judges' asses? These motherfuckers only give relief to those crooks, and the common people die, but justice is not served.</i>	0	<b>Abusive</b> ✓ Rationale: The sentence contains offensive language and derogatory terms, which are disrespectful and intended to harm or demean. The use of vulgar words like “gand” (ass) and “bhenchod” (sister-fucker).

Table 12: Examples showing zero-shot English prompt performance by the Llama-3-70B-Instruct model on the coarse-grained RUHSOLD dataset, with model justifications for categorizing specific samples as abusive or not. Sentences translated into English are displayed in italics.

Model Input	Ground	Prediction
randi olado tumne kab kab talvar chalai hai madorchod teri ma ki choot maru bosdike <i>When did you pick up the sword, you mother****er? I'll f*** your mother, you son of a b***h</i>	Abusive	The given sentence is classified as Abusive. The sentence contains offensive language and inappropriate content that is not suitable for general use. The words used in the sentence are vulgar, disrespectful, and highly offensive, making it unsuitable for any kind of communication. True. ✓
rt : bharwe masjid mae hoon mae <i>I am at the mosque, pimp</i>	Abusive	False. X The given sentence is not abusive. It appears to be a statement in the Hindi language, possibly expressing that someone is associated with a mosque.
raatein dengi bata neendo mein teri hi baat hai bhoolun kaisay tujhe tu toh khayalo mein sath hai <i>Nights tell me that it's only you in my dreams. How can I forget you? You are in my thoughts.</i>	Not Abusive	The given sentence can be classified as Abusive. The sentence contains aggressive and disrespectful language, such as "dengi" (money) and "bhoolun kaisay" (how to forget), which are commonly used in abusive or" X
i think base is akhrot halwa and topping of gulab jaman it is sweet dish <i>I think the base is walnut halwa and the topping is gulab jamun. It is a sweet dish.</i>	Not Abusive	The given sentence is not abusive. It appears to be a statement or description about food items, specifically mentioning "base is akhrot halwa" and "topping of gulab jaman," which are components of a sweet dish. True ✓
tu hijra hai...ae hijrechatya <i>You are a eunuch... hey you eunuch fellow</i>	Abusive	The sentence is considered abusive. The words used in this sentence are derogatory and offensive towards transgender individuals, which falls under the category of abusive language. It is important to treat others with respect. True ✓
ham to yah bhi jante hain ki yahoodi har chij ko dhundhte rahte hain aur yahoo bhaiyon ko vah chij nahin milati aur itni hi purani academy nabiyon ki dor ki ki yahan insan pahle raha karte the vahi chijen mil rahi hai jo chij inko chahie vah chij nahin mil rahi hai <i>We also know that Jews keep searching for everything, and those brothers cannot find the thing they want. The academy is as old as the prophets' era, where people used to live. They are getting those things, but not the one thing they need.</i>	Abusive	The given sentence can be classified as Abusive. The language used is aggressive, and there is an underlying tone of disrespect and hostility towards a specific group ("yahoodi"). The sentence also contains inappropriate religious references that may be considered offensive. True ✓

Table 13: Examples showing zero-shot English prompt performance by the Qwen-2-7B-Instruct model on the coarse-grained RUHSOLD dataset, with model justifications for categorizing specific samples as abusive or not. Sentences translated into English are displayed in italics.

Model Input	Ground	Prediction
abe akhrot si shakal ke #arhaankhan tu bolega #asimriaz ko shakal dek khud ko deka h itne bure to log adhar card m b ni dikhte #wesupportsidharthshukla <i>Hey, with a face like a walnut, #arhaankhan, you'll talk to #asimriaz? Look at yourself first. People don't even look this bad in their ID cards. #wesupportsidharthshukla</i>	0	<b>Abusive</b> ✓ Reasoning: The sentence contains abusive language and derogatory terms targeting an individual's physical appearance. It also includes a hashtag that supports another individual, potentially indicating bias and further emphasizing the abusive nature of the comment. Note: The language used
koi wja to ho gi ? <i>There must be some reason?</i>	1	<b>Not Abusive</b> ✓ Reasoning: The sentence koi wja to ho gi? is in Urdu and translates to 'There must be some reason?' in English. It doesn't contain any offensive language, threats, or derogatory terms, so it
aby han mn bhol gya ye to sasti randi hy <i>Oh yes, I forgot. She's just a cheap prostitute.</i>	0	<b>Abusive</b> ✓ Reasoning: The sentence contains derogatory language and insults, which are directed towards a person. The use of the term randi is particularly offensive and demeaning, making this sentence clearly abusive in nature. Additionally, the sentence
hum sy toh na ho paye ga <i>We won't be able to do this.</i>	1	<b>Not Abusive</b> ✓ Explanation: The sentence hum sy toh na ho paye ga is in Urdu language and it translates to 'We won't be able to do this.' There are no abusive words or phrases in this sentence.
bhenchod messi kahin se bhi goal daal deta hai <i>F***ing Messi can score a goal from anywhere.</i>	0	<b>Abusive</b> ✓ Explanation: The sentence contains a derogatory term in Hindi, which is considered vulgar and offensive. Therefore, it is classified as abusive. Additionally, the sentence also contains a reference to a person (Messi) which can be considered as
rt : logon ke relationship ka randi rona dekh dekh ke single hone pe proud feel hota hai. <i>RT: Watching people whine about their relationships makes me feel proud of being single.</i>	0	<b>Abusive</b> ✓ Reasoning: The sentence contains derogatory language and insults, referring to someone as a randi which is a highly offensive term in Hindi/Urdu. This kind of language is disrespectful and intended to demean and humiliate.
nhi bc apne sbhi londe lage hue hain <i>No, bro, all your boys are busy at it.</i>	0	<b>Abusive</b> ✓ The sentence is in Hindi and it translates to 'No, bro, all your boys are busy at it.' The use of the word londa (translated here as boys) is vulgar and offensive, making this sentence

Table 14: Examples showing zero-shot English prompt performance by the Qwen-2-72B-Instruct model on the coarse-grained RUHSOLD dataset, with model justifications for categorizing specific samples as abusive or not. Sentences translated into English are displayed in italics.

Model Input->Roman Urdu [Translation]	Ground	Llama-70B	Qwen-72B	Lughat-8B
khabasaat se bharpoor yaar maar ein tu....assi dost samjhday umar guzar deti tu yahoodi nikaleyay <i>You are filled with filth, man. We considered you a friend and spent            a lifetime with you, and it turns out you're a Jew.</i>	Religious hate	Religious hate ✓	Not abusive X	Religious hate ✓
apne gawah banna hay ya marriage hall ki booking krwani hay? krne deen jo krty han <i>Do you want to become a witness yourself or are you arranging a            marriage hall booking? Let people do what they are doing.</i>	Not Abusive	Not abusive ✓	Religious Hate X	Abusive X
aunty aap sunni shia barelvi ahmedi etc etc etc kya ho <i>Aunty, are you Sunni, Shia, Barelvi, Ahmadi, etc.?</i>	Religious hate	Not abusive X	Religious hate ✓	Religious hate ✓
ye koi randi khana he? <i>Is this some kind of brothel?</i>	Profane	Sexual hate X	Profane ✓	Sexual hate X
ap transgender ki representation kro gee kal mera jism meeri merzi march mn shamal ho k <i>You will represent transgender people by joining tomorrow's 'My            Body, My Choice' march.</i>	Sexual hate	Not abusive X	Abusive X	Sexual hate ✓
aur ye phattu unko appreciation dete hue <i>And this coward is giving them appreciation.</i>	Abusive	Not abusive X	Religious hate X	Abusive ✓
agar unmute na kre to bol dena chehre to langoor hai tera or tu hijra hai <i>If they don't unmute, then tell them their face looks like a langur            (monkey) and that they are a eunuch.</i>	Sexual hate	Abusive X	Religious hate X	Profane X

Table 15: Fine-grained classification prediction of RUHSOLD using few-shot Roman Urdu prompt by the Llama-3-8B-Instruct, Qwen-2-72B-Instruct, and Lughat-1.0-8B model.

```

Fluency Prompt

SYSTEM:
You are an expert in evaluating the fluency of Roman Urdu text.

INPUT:
Please evaluate the fluency of the following text given in Roman Urdu.

-----
### Instructions:
- Task: Evaluate the fluency of the text.
- Scoring: Provide a score from 1 to 5, where:
  - 5: The text is highly fluent, with no grammatical errors, unnatural wording, or stiff syntax.
  - 4: The text is mostly fluent, with minor errors that do not impede understanding.
  - 3: The text is moderately fluent, with noticeable errors that may slightly affect comprehension.
  - 2: The text has low fluency, with frequent errors that hinder understanding.
  - 1: The text is not fluent, with severe errors that make it difficult to understand.
- Explanation: Support your score with specific examples to justify your evaluation.

-----
### Output Format:
Return only the following JSON object, with no extra text:
"Fluency":
"Score": "<the score>",
"Explanation": "<your explanation on how you made the decision>"

-----
**Text to Evaluate:**
{sentence}

```

Figure 6: Prompt used with GPT-4 to find fluency.

## Content Prompt

```
SYSTEM:
You are an expert evaluator of finding text accuracy.

INPUT:
Please evaluate the accuracy of the following Roman Urdu text in <<target>> by comparing it to
the provided reference text.

-----
### Instructions:
- Task: Compare the text to the reference text.
- Identify Mistakes: List all mistakes related to accuracy.
  - Omission: Missing words, phrases, or information present in the reference text.
  - Addition: Extra words, phrases, or information not present in the reference text.
  - Others: Mistakes that are hard to define or categorize.

-----
### Output Format:
Return only the following JSON object:

"Accuracy":
"Mistakes":
<Mistake Types: summarize the mistake>

-----
Reference Text:
{reference}

Text to Evaluate:
{hypothesis}
```

Figure 7: Prompt used with GPT-4 to find cohesion errors—specifically lexical (LE) and grammatical (GE) errors.

## Cohesion Prompt

```
SYSTEM:
You are an expert evaluator of cohesion in text.

INPUT:
Please evaluate the cohesion of the following Roman Urdu text in <<target>> by comparing it to the
provided reference text.

-----
### Instructions:
- Task: Evaluate the cohesion of the text.
- Definition: Cohesion refers to how different parts of a text are connected using language structures
like grammar and vocabulary. It ensures that sentences flow smoothly and the text makes sense as a whole.
- Identify Mistakes: List all mistakes related to cohesion.
- Separate the mistakes into:
  - Lexical Cohesion Mistakes: Issues with vocabulary usage, incorrect or missing synonyms, or overuse
of certain words that disrupt the flow.
  - Grammatical Cohesion Mistakes: Problems with pronouns, conjunctions, or grammatical structures that
link sentences and clauses.
- Provide Lists: Provide separate lists for lexical cohesion mistakes and grammatical cohesion mistakes.
Provide empty lists if there are no mistakes.

-----
### Output Format:
Return only the following JSON object:

"Cohesion":
"Lexical Cohesion Mistakes": "<list mistakes here>"
"Grammatical Cohesion Mistakes": "<list mistakes here>"

-----
Reference Text:
{reference}

Text to Evaluate:
{hypothesis}
```

Figure 8: Prompt used with GPT-4 to find cohesion errors—specifically lexical (LE) and grammatical (GE) errors.