Overview of BLP-2025 Task 1: Bangla Hate Speech Identification

Md Arid Hasan[1], **Firoj Alam**[2], **Md Fahad Hossain**[3], **Usman Naseem**[4], **Syed Ishtiaque Ahmed**[1]
[1]University of Toronto, Canada, [2]Qatar Computing Research Institute, Qatar
[3]Daffodil International University, [4]Macquarie University, Australia
{arid, ishtiaque}@cs.toronto.edu, fialam@hbku.edu.qa
https://multihate.github.io

Abstract

Online discourse in Bangla is rife with nuanced toxicity expressed through code-mixing, dialectal variation, and euphemism. Effective moderation thus requires fine-grained detection of hate **type**, **target**, and **severity**, rather than a binary label. To address this, we organized the Bangla Hate Speech Identification Shared Task at **BLP 2025 workshop**, co-located with **IJCNLP-AACL 2025**, comprising three subtasks: (1A) hate-type detection, (1B) hate-target detection, and (1C) joint prediction of type, target, and severity in a multi-task setup. The subtasks attracted 161, 103, and 90 participants, with 36, 23, and 20 final submissions, respectively, while a total of 20 teams submitted system description papers. The submitted systems employed a wide range of approaches, ranging from classical machine learning to fine-tuned pretrained models and zero-/few-shot LLMs. We describe the task setup, datasets, and evaluation framework, and summarize participant systems. All datasets and evaluation scripts are publicly released.[1]

1 Introduction

Hate speech detection has emerged with the wide use of social media and online communication platforms, where users can rapidly share opinions, comments, and multimedia content. The proliferation of such content has led to an increase in harmful content (Walther, 2022). This has also facilitated the spread of hate speech language that targets individuals or groups based on attributes such as religion, ethnicity, gender, or political affiliation (Fortuna and Nunes, 2018). Therefore, detecting hate speech automatically is crucial for maintaining safe online environments and preventing real-world consequences such as discrimination and violence. While substantial progress has been made for English (Albladi et al., 2025) and other high-resource languages (Das et al., 2024), hate speech detection in Bangla remains a significant challenge due to the scarcity of annotated datasets and linguistic diversity (Sharma et al., 2025; Das et al., 2022b; Haider et al., 2025; Romim et al., 2022).

Early studies in Bangla hate speech detection focused on classical machine learning methods (Kiela et al., 2020; Mridha et al., 2021; Romim et al., 2022), deep learning models (Romim et al., 2022; Keya et al., 2023), and pretrained models that are primarily designed for English (Mridha et al., 2021). However, these methods struggle in understanding the deep cultural, social, and linguistic nuances that shape hate expression in Bangla (Al Maruf et al., 2024). These include context-sensitive slurs, metaphorical or sarcastic insults, and region-specific idiomatic phrases that are often misinterpreted or overlooked by standard models (Jahan et al., 2022). Recently, large language models (LLMs) such as GPT-5, Gemini (Comanici et al., 2025), Qwen (Yang et al., 2025), and Llama (Dubey et al., 2024) have achieved impressive generalization in NLP tasks but often underperform in hate speech detection for low-resource languages like Bangla. Their limited ability to grasp cultural nuances, implicit hate, and context-specific expressions (Zahid et al., 2025) underscores the need for domain adaptation and culturally aware training strategies.

Prior studies are limited to single-task datasets, focusing on only one dimension, such as hate type, which restricts the development of more sophisticated models capable of performing a multi-faceted analysis (e.g., simultaneously identifying a comment's type, severity, and target). Therefore, we emphasize community engagement and organized a shared task at BLP 2025[2] to address this challenge. The task consists of three subtasks[3] and

[1]https://github.com/AridHasan/blp25_task1

[2]https://blp-workshop.github.io/

[3]Subtask 1A: Identifying type of hate, Subtask 1B: Identifying target of hate, Subtask 1C: Identifying type, severity,

Proceedings of the Second Workshop on Bangla Language Processing (BLP-2025), pages 388–401

aims to foster the development of robust, multi-task Bangla hate speech detection systems by providing a large, carefully curated dataset spanning multiple domains. By encouraging diverse approaches ranging from classical models to LLMs and solutions, the shared task promotes collaboration and the exploration of various techniques tailored to the cultural, social, and linguistic nuances.

A total of 161, 103, and 89 teams registered for subtasks 1A, 1B, and 1C, respectively, with 37, 24, and 21 teams making official submissions on the test set. Among these, 19 teams also submitted system description papers. On the development-test set, there were 925 submissions from 56 teams for 1A, 312 submissions from 23 teams for 1B, and 224 submissions from 18 teams for 1C. For the final test set, the number of submissions were 421, 271, and 202 for subtasks 1A, 1B, and 1C, respectively.

2 Task and Dataset

2.1 Task

This shared task focused on detecting hate speech in Bangla, a low-resource language with rich morphology and regional dialects. To comprehensively address the complexity of hateful content, we divided the task into three subtasks: *(i)* identifying the *Type of Hate*, *(ii)* determining the *Target of Hate*, and *(iii)* a comprehensive multi-task setup that jointly predicts the *Type of Hate*, *Severity of Hate*, and *Target of Hate*. This design aimed to move beyond simple binary detection and encourage the development of models capable of addressing the nuanced linguistic and social dimensions of hate speech in Bangla. Task descriptions are provided below.

Subtask 1A: Type of Hate This subtask is defined as *"detect the type of hate that is expressed in the text"*. This is a multi-class classification task that requires determining whether a given instance falls into one of the following categories: *Abusive*, *Sexism*, *Religious Hate*, *Political Hate*, *Profane*, or *None*.

Subtask 1B: Target of Hate This subtask is defined as *"detect the target of hate that is expressed towards **whom** in the text"*. It is formulated as a multi-class classification problem, where the goal is to determine whether the hateful expression is aimed at *Individuals*, *Organizations*, *Communities*, *Society*, or *None*.

Subtask 1C: Multi-task Setup This subtask is designed as a multi-task learning setup that jointly addresses three dimensions of hate speech detection in Bangla: *Type of Hate*, *Severity of Hate*, and *Target of Hate*. Unlike the single-task formulations in Subtask 1A and Subtask 1B, this setup requires models to perform simultaneous predictions across all three aspects for each input text. The motivation behind this design is to encourage the development of models capable of capturing the inter-dependencies between different facets of hateful content, such as the way severity may vary depending on the type of hate or how certain targets are more likely to be associated with specific hate categories. Framing the problem as a multi-task learning challenge encourages the development of more robust and context-sensitive systems that go beyond individual classification tasks and better capture the complex, multidimensional characteristics of hate speech in Bangla. The target classes for *Type of Hate* and *Target of Hate* are the same as defined in Subtask 1A and Subtask 1B, respectively, while the *Severity of Hate* task categorizes instances into *Severe*, *Mild*, and *Little to None*. Here, *Severe* denotes strongly derogatory or inciteful content, *Mild* refers to moderately offensive or implicitly hateful expressions, and *Little to None* indicates content with minimal or no hateful intent.

2.2 Dataset

We utilized the *BanglaMultiHate* dataset (Hasan et al., 2025b) for this shared task. This dataset comprises comments from YouTube, sourced from the Somoy Bangla News channel.[4] This dataset covers 19 different topics, such as *Disaster, Entertainment, Health, Politics, Religion, Science, Sports*, etc. For this shared task, we utilize the training set as the official training set of the shared task. The development set was further divided into development and development-test[5] subsets using a stratified sampling approach to preserve class balance. Finally, the test is used for system evaluation and participant ranking. The detailed distribution of the data split is presented in Table 1.

and target of hate in multi-task setup

[4] https://www.youtube.com/@somoynews360

[5] This development test set is used as a test set for the development phase.

Class	Train	Dev	DT	Test	Total
Type of Hate					
Abusive	8,212	564	549	2,312	**11,637**
Political Hate	4,227	291	283	1,220	**6,021**
Profane	2,331	157	185	709	**3,382**
Religious Hate	676	38	40	179	**933**
Sexism	122	11	8	29	**170**
None	19,954	1,451	1,447	5,751	**28,603**
Total	**35,522**	**2,512**	**2,512**	**10,200**	**50,746**
Severity of Hate					
Litle to None	23,489	1,703	1,714	6,737	**33,643**
Mild	6,853	483	426	2,001	**9,763**
Severe	5,180	326	372	1,462	**7,340**
Total	**35,522**	**2,512**	**2,512**	**10,200**	**50,746**
Target of Hate					
Community	2,635	179	159	759	**3,732**
Individual	5,646	364	391	1,571	**7,972**
Organization	3,846	292	292	1,152	**5,582**
Society	2,205	141	142	625	**3,113**
None	21,190	1,536	1,528	6,093	**30,347**
Total	**35,522**	**2,512**	**2,512**	**10,200**	**50,746**

Table 1: Class label distribution of the shared task dataset. DT: development-test.

Annotation and Annotators Agreement The annotation of the *BanglaMultiHate* dataset (Hasan et al., 2025b) was conducted by a trained team of 35 native Bangla-speaking undergraduate students, with each comment labeled independently by three annotators and finalized by majority vote or consensus when needed. Quality checks and supervision ensured consistent standards. Inter-annotator agreement, measured using Fleiss' Kappa, showed substantial to almost perfect agreement across tasks, with scores of 0.71 for type of hate, 0.84 for severity of hate, and 0.79 for target of hate, while more fine-grained tasks yielded slightly lower agreement due to increased complexity.

3 Evaluation Framework

3.1 Evaluation Measures

We used the *unweighted Micro-F1 score* as the evaluation metric for Subtask 1A and 1B, while weighted Micro-F1 score is used for Subtask 1C, with the corresponding datasets and evaluation scripts made publicly accessible online.[6] To establish reference points, we included the majority and random baselines along with the n-gram. The majority baseline predicts the most frequent class in the training data for every instance in the test set, while the random baseline assigns classes to test instances uniformly at random. We also provide a simple n-gram ($n = 1$) baseline using 5,000 features, with a linear SVM implemented to capture surface-level lexical patterns.

3.2 Task Organization

For the shared task, we released four datasets: the training set, development set, development-test set, and test set for each subtask, as summarized in Table 1. The development set was intended for hyperparameter tuning, while the development-test set was provided without labels to enable participants to assess their systems during the development phase. The test set was utilized for the final evaluation and ranking of submissions. All the subtasks (Subtask 1A,[7] Subtask 1B,[8] and Subtask 1C[9]) of this shared task was conducted in two phases, with the submission platform hosted on CodaBench.

Development Phase During this phase, participants were provided with the training set, development set, and development-test set. The development-test set was released without gold labels to ensure fair competition. This design encouraged participants to iteratively refine and optimize their models using the labeled training and development sets, while evaluating their systems on the unlabeled development-test set. A live leaderboard was made available throughout this phase, allowing teams to monitor the relative performance of their submissions in real time and to benchmark their approaches against other participants. This competitive setup fostered active engagement and provided valuable insights into the effectiveness of different modeling strategies prior to the final evaluation stage.

Evaluation Phase In this phase, the test set was released without gold labels, and participants were allotted a eight-day window to submit their final predictions. The test set served as the basis for the official evaluation and ranking of systems. To preserve fairness and prevent overfitting to the test data, the leaderboard was kept private during this phase. While participants were permitted to submit multiple systems (per day 100 submissions and in total 1000 submissions), the corresponding evaluation scores were withheld from them. For the

[6] https://github.com/AridHasan/blp25_task1

[7] https://www.codabench.org/competitions/9559/

[8] https://www.codabench.org/competitions/9560/

[9] https://www.codabench.org/competitions/9561/

final ranking, only the last valid submission from each team was considered, ensuring consistency and comparability across participants.

The test set along with its gold labels was released after the competition concluded, allowing participants to perform additional experiments, conduct error analyses, and further refine their models.

4 Results and Overview of the Systems

4.1 Results

In this section, we present the outcomes of the shared task across both phases. Overall, participation was strong, with 56, 23, and 18 teams submitting systems during the development phase and 36, 23, and 18 teams in the evaluation phase for Subtask 1A, 1B, and 1C, respectively. Tables 2, 4, and 5 report the performance of all submitted systems on the development-test and test sets, alongside the majority and random baselines for comparison for Subtask 1A, 1B, and 1C, respectively. The official ranking was determined based on results from the test set. Notably, some teams participated only in the development phase but not in the evaluation phase, and vice versa, as indicated by ✗.

A comparison of results across the development-test and test sets indicates that performance differences among teams were minimal across three subtasks. This suggests that the models generally did not exhibit overfitting to the development-test set. In several instances, the systems even achieved higher performance on the test set than on the development-test set, highlighting the robustness and generalizability of the submitted approaches.

Table 3 provides a comprehensive overview of the approaches employed by participating teams across the three subtasks. The majority of teams relied on transformer-based architectures, particularly BanglaBERT, XLM-RoBERTa, and MuRIL, reflecting their effectiveness for Bangla and low-resource contexts. Several systems integrated ensemble strategies, combining multiple fine-tuned models to enhance robustness and performance. Moreover, we observed that systems employing ensemble techniques achieved the highest rankings on the leaderboard on all three subtasks. A smaller subset of teams experimented with classical machine learning and neural network baselines, often as complementary or comparative models. Additionally, data preprocessing and data augmentation were common practices to improve text quality and address class imbalance. A few teams further adopted LLMs (such as GPT-4.1, Llama3, Gemma 2, and Qwen3) and few-shot prompting approaches (e.g., Qwen-based systems), showcasing an emerging shift toward generative and low-resource adaptive methods.

4.2 Overview of the Systems

We summarize each participating system and its corresponding ranking on the leaderboard below.

Code_Gen (Islam et al., 2025) achieved the best performance in subtask 1A, ranked 2^{nd} and 3^{rd} for subtask 1B and subtask 1C, fine-tuned BanglaBERT (Bhattacharjee et al., 2021), multilingual E5 (Wang et al., 2024), MuRIL (Khanuja et al., 2021), XLM-RoBERTa (Conneau et al., 2020), and DistilBERT using token-aware adversarial contrastive training and layer-wise learning rate decay to enhance optimization and stability. Initially, the authors preprocessed through normalization, cleaning, and tokenization, and then incorporated data augmentation with a 70:30 train–validation split. Moreover, the authors utilized individual model logits that were generated and subsequently ensembled through different combinations of models to improve predictive performance.

SyntaxMind (Riad, 2025) integrates contextual language representations with sequential and local feature extraction mechanisms to enhance the classification task. To generate contextual embeddings, the authors used BanglaBERT encoder, which is then processed in parallel through a CNN that captures local n-gram patterns through multiple kernel sizes, and a GRU utilizes sequential dependencies with bidirectional recurrence. Moreover, both CNN and GRU employ self-attention, and the outputs of the CNN and GRU attentions are then fused through a dense layer. This team ranked 2^{nd} and 5^{th} in subtask 1A and 1B, respectively.

TeamHateMate (Hasan and Mahim, 2025) fine-tuned BanglaBERT using a two-stage cascading architecture for all three sub-tasks: a binary classifier to separate hate from non-hate, followed by a multi-class classifier for fine-grained categorization. Each stage was optimized through k-fold cross-validation and ensembled through majority voting. The authors also incorporated attention pooling, dropout, and hidden layers to enhance performance and tuned hyperparameters separately for each subtask. Their system ranked 4^{th} in subtask 1A, while ranked 1^{st} in both subtask 1B and 1C. The authors further attempted data augmentation via back translation and class balancing; however,

R.	Team Name	Dev P.	Eval. P.
1	Code_Gen (Islam et al., 2025)	0.7580	0.7362
2	SyntaxMind (Riad, 2025)	0.7440	0.7345
3	zannatul_007	0.7440	0.7340
4	TeamHateMate (Hasan and Mahim, 2025)	0.7544	0.7331
5	Ecstasy (Hasan et al., 2025a)	0.7564	0.7328
6	Gradient Masters (Rahman et al., 2025b)	0.7488	0.7323
7	Catalyst (Hasan and Hasan, 2025)	0.7572	0.7305
8	BElite (Tripty et al., 2025b)	0.7444	0.7282
9	Retriv (Saha et al., 2025)	0.7572	0.7275
10	CoU-CU-DSG (Alam et al., 2025)	0.7217	0.7273
11	CUET-NLP_Zenith (Hossan et al., 2025)	0.7357	0.7263
12	NSU_MILab (Rahman et al., 2025a)	0.7138	0.7250
13	abid_al_hossain	0.7416	0.7238
14	PentaML (Tahmid et al., 2025)	0.7162	0.7178
15	HateNetBN (Anam and Mazumder, 2025)	0.7365	0.7133
16	Computational StoryLab (Prama et al., 2025)	0.7404	0.7111
17	minjacodes9	0.5852	0.7075
18	Heisenberg (Yasir, 2025)	0.7086	0.7070
19	pritampal98	0.7373	0.7057
20	Bahash-AI (Laskar and Paul, 2025)	✗	0.7028
21	Velora (Sayem and Rahman, 2025)	0.7197	0.7013
22	fatin_anif	✗	0.6954
23	PerceptionLab (Fahim and Khan, 2025)	0.6584	0.6941
24	adriti12	0.3264	0.6921
25	nuralflow	0.7038	0.6901
26	Team_NSU_Strugglers	0.6899	0.6871
27	CUET_Sntx_Srfrs (Tripty et al., 2025a)	✗	0.6867
28	abir_bot69	0.6393	0.6840
29	antara_n_15	0.6899	0.6815
30	PromptGuard (Hossan and Roy Dipta, 2025)	0.6879	0.6761
31	quasar	0.1075	0.6733
32	shahriar_9472	0.6720	0.6689
33	intfloat	0.6712	0.6634
34	naim-parvez	✗	0.6587
35	Baseline (Majority)	0.5760	0.5638
36	teddymas	✗	0.4589
37	Baseline (Random)	0.1465	0.1638
38	mizba	0.7237	0.1077
–	messalmonem	0.7588	✗
–	nur_163	0.7568	✗
–	cuet_1376	0.7393	✗
–	manik	0.7361	✗
–	Tensoryus	0.7357	✗
–	phantom_troupe	0.7345	✗
–	hasnat	0.7329	✗
–	rashfi_21	0.7325	✗
–	Md.Fahad Ali	0.7313	✗
–	rabeya_akter	0.7237	✗
–	foysal_ahmed	0.7197	✗
–	md_abdur_rahman	0.7166	✗
–	no_name	0.7102	✗
–	saminyasir007	0.7062	✗
–	shuvodwip_saha	0.7030	✗
–	mhd88	0.6979	✗
–	tesnik	0.6883	✗
–	walisa_alam	0.6879	✗
–	deleted_user_29306	0.6815	✗
–	rakib_hossan	0.6620	✗
–	zulkarnyn420	0.6357	✗
–	loser1	0.5760	✗
–	unknown333	0.5760	✗
–	deleted_user_31920	0.3332	✗
–	rahi_12	0.1210	✗

Table 2: Official ranking of the subtask 1A on the test set. – only participated in the Development Phase. ✗ indicates team has not submitted system in the respective phase. R.: Rank, Dev P.: Development Phase, Eval. P.: Evaluation Phase.

both approaches failed to yield further gains.

Ecstasy (Hasan et al., 2025a) conducted a detailed linguistic analysis of 35,522 Bangla hate speech samples using TF-IDF to identify distinctive lexical patterns for each hate category. Category-specific keywords were embedded into model prompts to provide contextual cues. The model was fine-tuned using LoRA adapters ($r = 64$, $\alpha = 128$) on Llama-3.1-8B with optimized hyperparameters for efficiency. Incorporating keyword-based prompts notably enhanced the model's ability to capture culturally nuanced hate speech patterns unique to Bangla. This team ranked 5^{th} and and 4^{th} in subtask 1A and 1C, respectively.

Gradient Masters (Rahman et al., 2025b) began with BiLSTM and LSTM with attention using pre-trained Bangla embeddings; however, the performance of RNN models prompted a shift to transformer-based models. Their main pipeline fine-tuned BanglaBERT, MuRIL, XLM-R, and DistilBERT with custom classification heads, with BanglaBERT performing best due to language-specific pretraining. To handle severe class imbalance, the authors applied stratified k-fold cross-validation, text normalization, and adversarial training (FGSM). Ensembles of best performing models per subtask were used for final predictions, without post-processing. This team ranked 6^{th} and 3^{rd} in subtask 1A and 1C, respectively.

Catalyst (Hasan and Hasan, 2025) fine-tuned pretrained models such as XLM-RoBERTa (Conneau et al., 2020), mDeBERTa-v3, MuRIL (Khanuja et al., 2021), and IndicBERTv2 (Doddapaneni et al., 2022), optimized using AdamW, mixed-precision training, and task-specific hyperparameters. For single-task setups (e.g., subtask 1A, subtask 1B), authors combined multiple models through hard-voting ensembles to enhance robustness and generalization. For the subtask 1C (multi-task), this system implemented a shared transformer encoder with three task-specific classification heads to jointly predict hate type, severity, and target. Across all subtasks, authors found that multilingual pre-trained transformers and ensembling provided consistent improvements in model stability and performance. This team ranked 7^{th}, 8^{th}, and 10^{th} in subtask 1A, subtask 1B, and subtask 1C, respectively.

BElite (Tripty et al., 2025b) fine-tuned BanglaBERT, mBERT, and XLM-RoBERTa on the dataset and then created an ensemble of these models. Two ensemble strategies were applied: simple averaging and a weighted ensemble, where the weights of individual models were determined based on their weighted F1 scores on the validation

Team	Model													Misc.	
	Classical Model	Neural Networks	BanglaBERT	XLM-RoBERTa	MuRIL	FGSM	IndicBERT	DistilBERT	mE5	mDeBERTa	mBERT	LLMs	Few Shot	Ensemble	Data Preprocessing	Data Augmentation
Code_Gen (Islam et al., 2025)			✓	✓	✓			✓	✓					✓	✓	✓
SyntaxMind (Riad, 2025)		✓	✓									✓		✓	✓	
Ecstasy (Hasan et al., 2025a)		✓	✓	✓	✓			✓								
Gradient Masters (Rahman et al., 2025b)			✓	✓	✓	✓								✓	✓	
Catalyst (Hasan and Hasan, 2025)				✓	✓		✓			✓				✓	✓	
BElite (Tripty et al., 2025b)			✓	✓							✓			✓	✓	
Retriv (Saha et al., 2025)		✓	✓		✓		✓							✓		
CoU-CU-DSG (Alam et al., 2025)			✓	✓	✓											
CUET-NLP_Zenith (Hossan et al., 2025)			✓	✓										✓		
NSU_MILAB (Rahman et al., 2025a)			✓	✓	✓		✓							✓		
PentaML (Tahmid et al., 2025)		✓	✓	✓			✓									
HateNetBN (Anam and Mazumder, 2025)			✓	✓												
Computational StoryLab (Prama et al., 2025)			✓	✓			✓				✓	✓				
Heisenberg (Yasir, 2025)			✓					✓							✓	✓
Bahash-AI (Laskar and Paul, 2025)			✓												✓	✓
Velora (Sayem and Rahman, 2025)			✓												✓	
PerceptionLab (Fahim and Khan, 2025)			✓													✓
PromptGuard (Hossan and Roy Dipta, 2025)												✓	✓		✓	✓
CUET_Sntx_Srfrs (Tripty et al., 2025a)	✓														✓	✓

Table 3: Overview of the approaches used in the submitted systems across three subtasks.

set. The results show that the weighted ensemble outperformed all individual models as well as the simple averaging approach. This team ranked 8^th, 9^{th}, and 5^{th} in subtask 1A, 1B, and 1C, respectively.

Retriv (Saha et al., 2025) employed soft-voting ensembles of transformer models, such as MuRIL, BanglaBERT, and IndicBERTv2, for subtasks 1A and 1B to enhance predictive stability, and a MuRIL-based multi-task framework for subtask 1C to jointly optimize related objectives with inputs truncated to 128 tokens and tuned hyperparameters ($lr = 2e^{-5}$, batch size 16, 3 epochs) applied uniformly. Authors further experimented with hybrid transformer–RNN architectures (BiLSTM, BiGRU) as classification heads to capture sequential context. This team ranked 9^{th}, 10^{th}, and 7^{th} in subtask 1A, 1B, and 1C, respectively.

CoU-CU-DSG (Alam et al., 2025) utilized a weighted probabilistic fusion framework that leverages multiple transformer-based language models for the detection of Bangla hate speech. This approach integrates BanglaBERT, XLM-RoBERTa, and MuRIL, combining their output probabilities through a weighted fusion strategy to leverage the complementary strengths of Bangla-specific and multilingual models. The output of BanglaBERT outperforms other models. This team ranked 10^{th} and 15^{th} in subtask 1A and 1B, respectively.

CUET-NLP_Zenith (Hossan et al., 2025) empliyed a multi-task architecture for Bangla hate speech detection, leveraging a shared transformer backbone with an ensemble of pre-trained models, such as BanglaBERT[10], XLM-RoBERTa, and BanglaBERT (Bhattacharjee et al., 2021). The system jointly classifies hate type, severity, and target group using shared contextual embeddings from the transformer encoder, where text is tokenized to 128 tokens per sequence and processed into 768-dimensional embeddings, followed by [CLS] pooling and dropout regularization. Moreover, task-specific learning rates, a linear scheduler, and summed cross-entropy loss were utilized to fine-tune the model. This team ranked 11^{th}, 13^{th} in subtask 1A and 1B, respectively.

NSU_MILab (Rahman et al., 2025a) evaluated four transformer models, such as BanglaBERT, XLM-RoBERTa, IndicBERT, and Bengali Abusive

[10] https://huggingface.co/sagorsarker/bangla-bert-base

MuRIL, for Bangla hate speech detection. To improve robustness, we applied an ensemble strategy that averaged output probabilities across models, yielding consistent gains over individual systems. Post-competition refinements further confirmed the effectiveness of our ensemble approach in improving overall performance. This team ranked 12^{th} and 17^{th} in subtask 1A and 1B, respectively.

PentaML (Tahmid et al., 2025) fine-tuned multiple pre-trained BERT-based transformer models to classify hate speech in Bangla the comments. To improve performance, PentaML team applied linear probing on three fine-tuned models, allowing better use of learned representations. This lightweight approach achieved consistently better results across all subtasks. This team ranked 14^{th}, 11^{th}, and 13^{th} in subtask 1A, 1B, and 1C, respectively.

HateNetBN (Anam and Mazumder, 2025) utilized parameter-efficient architecture that leverages hierarchical representations from pre-trained transformer models by freezing the backbone to reduce computational cost. A layer-wise attention mechanism learns the relative importance of transformer layers, generating and aggregating context vectors for classification. This design enables effective integration of syntactic and semantic features, providing a lightweight yet powerful alternative to full fine-tuning for Bangla hate speech detection. This team ranked 15^{th} and 12^{th} in subtask 1A and 1B, respectively.

Computational StoryLab (Prama et al., 2025) utilized a multi-task framework built on transformer-based models like BanglaBERT, mBERT, and XLM-RoBERTa. The system uses four separate BERT-based models: a binary classifier to detect toxic comments, and three multiclass classifiers to predict hate type, severity, and target group. Each model includes a dropout layer and a linear output layer, with input processed as standard BERT inputs. Training employed categorical cross-entropy and BCEWithLogitsLoss, optimized with AdamW, while monitoring training and validation accuracy. This team ranked 16^{th} in both subtask 1A and 1B and 11^{th} in subtask 1C.

Heisenberg (Yasir, 2025) tokenized training data using the Bangla basic tokenizer, with stopwords removed. Dataset augmentation included 4,000 newly collected YouTube comments, synonym-based replacement on 8,000 samples, and back-translation of 27,000 samples. This system fine-tuned transformer-based models, including DistillBERT, BanglaHateBERT, BanglaT5, and BanglaBERT. This team ranked 18^{th} in subtask 1A.

Bahash-AI (Laskar and Paul, 2025) used BanglaBERT for all subtasks, applying minimal preprocessing. Subtasks 1A and 1B involved single-label classification, while subtask 1C used a multi-output setup with one-hot encoding and a combined loss to optimize all three labels simultaneously. To increase training size, Bangla texts were translated to English, paraphrased with $pegasus\_paraphrase$, and back-translated, adding 28,220 instances. Models were trained with batch size 16 and dropout 0.1, for 10 epochs with early stopping, and evaluated using F1-score. This team ranked 20^{th} and 17^{th} in subtask 1A and 1B, respectively.

Velora (Sayem and Rahman, 2025) fine-tuned BanglaBERT on a merged dataset combining the competition data with a publicly available Bangla hate speech corpus. To address class imbalance, this system applied back-translation augmentation, logit-adjusted loss, and CB-Focal loss, along with Bangla-specific preprocessing such as NFKC normalization and URL/punctuation removal. Training used a learning rate of 2e-5 (base) and 2e-4 (head), batch size 16, 12 epochs, and early stopping. This team ranked 21^{st} in subtask 1A.

PerceptionLab (Fahim and Khan, 2025) combined Domain-Adaptive Pretraining (DAPT) and multilingual transformers with supervised fine-tuning for hate speech classification. This system augmented the dataset with external corpora and curated examples to address class imbalance. Single-shot six-way classification outperformed hierarchical setups, and DAPT consistently improved performance, especially for majority classes. This team ranked 23^{rd} and 18^{th} in subtask 1A and 1B, respectively.

PromptGuard (Hossan and Roy Dipta, 2025) utilized a few-shot learning approach for Bangla hate speech detection, coordinated by a manager agent. For each input sentence, it generates a few-shot prompt enriched with examples from all six hate categories and category-specific keywords identified via chi-square correlation with the labels. Classification occurs over multiple "turns", with each turn sampling a new set of examples to create a fresh prompt for inference. The final label is determined by majority voting across turns, with additional iterations used to break ties. This team ranked 30^{th} in the leaderboard of subtask 1A.

CUET_Sntx_Srfrs (Tripty et al., 2025a) evaluated classical machine learning models (LR, DT, MNB, SVC, RF, KNN) using simple and hierarchical pipelines with preprocessing, n-gram features (TF-IDF and Count), and ensemble voting. Hierarchical classification combined with TF-IDF and majority-voting ensembles improved minority class detection while maintaining strong overall performance. This system also assessed the impact of preprocessing and n-gram choices, providing reproducible baselines for Bangla hate speech detection. This team ranked 21^{st} and 18^{th} in subtask 1B and 1C, respectively.

5 Related Work

Detecting offensive language and hate speech has become increasingly crucial with the rapid growth of social media, where harmful content spreads at scale (Jiang and Zubiaga, 2024; Sharma et al., 2022; Alam et al., 2022). Over the past decade, the field has seen a rapid methodological shift (Fortuna and Nunes, 2018), moving from lexicon-based techniques (Waseem and Hovy, 2016) and classical machine learning models (e.g., logistic regression, SVM, etc.) (Davidson et al., 2017) to transformer models and more recently to large language models (LLMs) (Albladi et al., 2025; Hasan et al., 2024).

Early approaches were primarily lexicon-based or relied on shallow statistical models, such as n-gram features with linear classifiers. Subsequent work advanced to deep learning architectures, including recurrent neural networks (e.g., LSTM), and more recently to transformer-based models such as BERT, XLM-R, MuRIL, and AraBERT. Transformer-based models consistently outperform traditional classifiers in the detection of offensive and hate speech (Sharif et al., 2021). Prior work has explored multi-task learning in Arabic (Djandji et al., 2020), code-mixed texts in Dravidian languages (B and A, 2021), and cross-lingual transfer with mBERT and LASER (Pelicon et al., 2021), although cultural biases remain a key limitation (Saumya et al., 2021a).

Kiela et al. (2020) evaluated hateful content detection using SVM, CNN, and LSTM. Multi-label hate speech detection has employed classical models and transformation-based methods (Ibrohim and Budi, 2019), while Mridha et al. (2021) proposed L-Boost, combining BERT embeddings with LSTM for Bangla and Banglish social media. Comparisons on Bangla YouTube and Facebook comments show that SVM often outperforms LSTM and Bi-LSTM (Romim et al., 2021). Hybrid BERT-GRU models have also been applied (Keya et al., 2023), and recent work emphasizes the detection of explainable hate speech (Yang et al., 2023; Piot and Parapar, 2025; Sariyanto et al., 2025).

Several datasets have been developed for hate and offensive content detection. The study of Gupta et al. (2022) created a 150K-comment dataset for Indic languages, and Sharif et al. (2021) studied multilingual code-mixed text, providing baselines for Dravidian languages (Saumya et al., 2021b; Chakravarthi et al., 2022). For Bangla, resources include labeled tweets and comments ranging from 3K to 50K examples (Das et al., 2022a; Romim et al., 2021; Sazzed, 2021; Romim et al., 2022; Das et al., 2022b), while Haider et al. (2024) introduced a multi-label transliterated dataset using LLM-based translation prompting.

Though research on hate speech detection has grown rapidly, deploying these systems in real-world applications remains challenging due to performance gap, cross-lingual transfer, and cultural biases. This shared task aims to advance research through community collaboration and a standardized evaluation framework. As an initial focus, we classified Bangla text samples according to three annotation tasks: Type of Hate, Severity of Hate, and Target of Hate. This framework provides a foundation for future studies, including multi-task learning and explainable hate speech detection.

6 Conclusion

We presented an overview of the Hate Speech Detection shared task at BLP 2025, which focused on identifying hate speech in Bangla social media text across multiple subtasks. Participating systems leveraged transformer-based models, with BanglaBERT, XLM-RoBERTa, and MuRIL being the most widely used, often combined in ensemble setups. Several teams also explored innovative strategies such as few-shot learning, adversarial training, and task-specific loss functions to improve classification. In general, the submissions demonstrated a mix of classical machine learning, neural networks, pretrained language models, and LLMs approaches. For future work, we aim to expand the task to multi-label and multi-modal hate speech detection, as well as develop more robust techniques to handle class imbalance and cultural nuances in Bangla text.

Limitation

The BLP-2025 hate speech detection shared task primarily targeted comment-level classification, which often overlooks broader contextual information. As a result, the identification of nuanced aspects such as hate type, severity, and target remains limited. Additionally, this edition focused exclusively on unimodal (text-only) models, leaving the exploration of multimodal approaches for future research.

Ethics and Broader Impact

The *BanglaMultiHate* dataset contains only textual comments and excludes any personally identifiable information, ensuring that it does not pose direct privacy concerns. However, because annotation is a subjective process, it may still introduce certain biases. To mitigate this risk, a well-defined annotation framework was employed along with comprehensive guidelines to promote consistency and reliability. However, we advise researchers and practitioners to remain aware of these inherent limitations when employing the dataset for modeling or further research.

References

Abdullah Al Maruf, Ahmad Jainul Abidin, Md Mahmudul Haque, Zakaria Masud Jiyad, Aditi Golder, Raaid Alubady, and Zeyar Aung. 2024. Hate speech detection in the bengali language: a comprehensive survey. *Journal of Big Data*, 11(1):97.

Ashraful Alam, Abdul Aziz, and Abu Nowshed Chy. 2025. Cou-cu-dsg at BLP-2025 Task 1: Leveraging weighted probabilistic fusion of language models for bangla hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Firoj Alam, Stefano Cresci, Tanmoy Chakraborty, Fabrizio Silvestri, Dimiter Dimitrov, Giovanni Da San Martino, Shaden Shaar, Hamed Firooz, and Preslav Nakov. 2022. A survey on multimodal disinformation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6625–6643, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Aish Albladi, Minarul Islam, Amit Das, Maryam Bigonah, Zheng Zhang, Fatemeh Jamshidi, Mostafa Rahgouy, Nilanjana Raychawdhary, Daniela Marghitu, and Cheryl Seals. 2025. Hate speech detection using large language models: A comprehensive review. *IEEE Access*.

Mohaymen Ul Anam and Akm Moshiur Rahman Mazumder. 2025. Hatenet-bn at BLP-2025 Task 1: A hierarchical attention approach for bangla hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Bharath B and S. Ajith A. 2021. Ssncse_nlp@dravidianlangtech-eacl2021: Offensive language identification on multilingual code mixing text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, DravidianLangTech*, pages 313–318. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin, Md Saiful Islam, Anindya Iqbal, M Sohel Rahman, and Rifat Shahriyar. 2021. Banglabert: Language model pretraining and benchmarks for low-resource language understanding evaluation in bangla. *arXiv preprint arXiv:2101.00204*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P McCrae. 2022. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, 56(3):765–806.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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.

Manjira Das, Sourabh Banerjee, Pushpak Saha, and Animesh Mukherjee. 2022a. Hate speech and offensive language detection in bengali. In *Proceedings of the 4th International Conference on Computational Linguistics (ACLing 2022)*. ArXiv preprint arXiv:2210.03479.

Mithun Das, Somnath Banerjee, Punyajoy Saha, and Animesh Mukherjee. 2022b. Hate speech and offensive language detection in Bengali. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 286–296, Online only. Association for Computational Linguistics.

Susmita Das, Arpita Dutta, Kingshuk Roy, Abir Mondal, and Arnab Mukhopadhyay. 2024. A survey on automatic online hate speech detection in low-resource languages. *arXiv preprint arXiv:2411.19017*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11 of *AAAI '17*.

Mouadh Djandji, Freddy Baly, Wissam Antoun, and Hady Hajj. 2020. Multi-task learning using arabert for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection (OSACT4)*, pages 97–101. European Language Resource Association.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *arXiv preprint arXiv:2212.05409*.

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 e-prints*, pages arXiv–2407.

Tamjid Hasan Fahim and Kaif Ahmed Khan. 2025. Perceptionlab at BLP-2025 Task 1: Domain-adapted bert for bangla hate speech detection: Contrasting single-shot and hierarchical multiclass classification. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

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

Vikram Gupta, Sumegh Roychowdhury, Mithun Das, Somnath Banerjee, Punyajoy Saha, Binny Mathew, Hastagiri Prakash Vanchinathan, and Animesh Mukherjee. 2022. Macd: Multilingual abusive comment detection at scale for indic languages. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks*.

Fabiha Haider, Fariha Tanjim Shifat, Md Farhan Ishmam, Deeparghya Dutta Barua, Md Sakib Ul Rahman Sourove, Md Fahim, and Md Farhad Alam. 2024. Banth: A multi-label hate speech detection dataset for transliterated bangla. *arXiv preprint arXiv:2410.13281*.

Fabiha Haider, Fariha Tanjim Shifat, Md Farhan Ishmam, Md Sakib Ul Rahman Sourove, Deeparghya Dutta Barua, Md Fahim, and Md Farhad Alam Bhuiyan. 2025. BanTH: A multi-label hate speech detection dataset for transliterated Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7217–7236, Albuquerque, New Mexico. Association for Computational Linguistics.

Kazi Reyazul Hasan, Mubasshira Musarrat, and Muhammad Abdullah Adnan. 2025a. Ecstasy at BLP-2025 Task 1: Tf-idf informed prompt engineering with lora fine-tuning for bangla hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md Arid Hasan, Firoj Alam, Md Fahad Hossain, Usman Naseem, and Syed Ishtiaque Ahmed. 2025b. Llm-based multi-task bangla hate speech detection: Type, severity, and target. *arXiv preprint arXiv:2510.01995*.

Md. Arid Hasan, Shudipta Das, Afiyat Anjum, Firoj Alam, Anika Anjum, Avijit Sarker, and Sheak Rashed Haider Noori. 2024. Zero- and few-shot prompting with LLMs: A comparative study with fine-tuned models for Bangla sentiment analysis. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17808–17818, Torino, Italia. ELRA and ICCL.

Mehedi Hasan and Mahbub Islam Mahim. 2025. Teamhatemate at BLP-2025 Task 1: Divide and conquer: A two-stage cascaded framework with k-fold ensembling for multi-label bangla hate speech classification. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Nahid Hasan and Nahid Hasan. 2025. Catalyst at BLP-2025 Task 1: Transformer ensembles and multi-task learning approaches for bangla hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md. Refaj Hossan, Kawsar Ahmed, and Mohammed Moshiul Hoque. 2025. Cuet-nlp_zenith at BLP-2025 Task 1: A multi-task ensemble approach for detecting hate speech in bengali youtube comments. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Rakib Hossan and Shubhashis Roy Dipta. 2025. Promptguard at BLP-2025 Task 1: A few-shot classification framework using majority voting and keyword similarity for bengali hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Muhammad Okky Ibrohim and Indra Budi. 2019. Multi-label hate speech and abusive language detection in Indonesian Twitter. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 46–57, Florence, Italy. Association for Computational Linguistics.

Shifat Islam, Emon Ghosh, and Abhishek Agarwala. 2025. Code_gen at BLP-2025 Task 1: Enhancing bangla hate speech detection with transformers through token-aware adversarial contrastive training and layer-wise learning rate decay. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md Saroar Jahan, Mainul Haque, Nabil Arhab, and Mourad Oussalah. 2022. BanglaHateBERT: BERT for abusive language detection in Bengali. In *Proceedings of the Second International Workshop on Resources and Techniques for User Information in Abusive Language Analysis*, pages 8–15, Marseille, France. European Language Resources Association.

Aiqi Jiang and Arkaitz Zubiaga. 2024. Cross-lingual offensive language detection: A systematic review of datasets, transfer approaches and challenges. *arXiv preprint arXiv:2401.09244*.

A. J. Keya, M. M. Kabir, N. J. Shammey, M. F. Mridha, M. R. Islam, and Y. Watanobe. 2023. G-bert: An efficient method for identifying hate speech in bengali texts on social media. *IEEE Access*, 11:79697–79709.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, and 1 others. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems (NeurIPS) 33*.

Sahinur Rahman Laskar and Bishwaraj Paul. 2025. Bahash-ai at BLP-2025 Task 1: Bangla hate speech detection using data augmentation and pre-trained model. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md. Firoj Mridha, Md. Abul Hasnat Wadud, Md. Abdul Hamid, Md. Mostafa Monowar, Md. Abdullah-Al-Wadud, and Atif Alamri. 2021. L-boost: Identifying offensive texts from social media post in bengali. *IEEE Access*, 9:164681–164699.

Anze Pelicon, Raghav Shekhar, Matjaz Martinc, Blaž Škrlj, Matthew Purver, and Simon Pollak. 2021. Zero-shot cross-lingual content filtering: Offensive language and hate speech detection. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 30–34. Association for Computational Linguistics.

Paloma Piot and Javier Parapar. 2025. Towards efficient and explainable hate speech detection via model distillation. In *European Conference on Information Retrieval*, pages 376–392. Springer.

Tabia Tanzin Prama, Christopher M. Danforth, and Peter Sheridan Dodds. 2025. Computational storylab at BLP-2025 Task 1: Hatesense: A transformer-based multi-task learning framework for comprehensive hate speech identification. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md. Mohibur Nabil Rahman, Muhammad Rafsan Kabir, Rakibul Islam, Fuad Rahman, Nabeel Mohammed, and Shafin Rahman. 2025a. Nsu_milab at BLP-2025 Task 1: Decoding bangla hate speech: Fine-grained type and target detection via transformer ensembles. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Naimur Rahman, Md Sakhawat Hossain, and Syed Mohaiminul Hoque. 2025b. Gradient masters at BLP-2025 Task 1: Advancing low-resource nlp for bengali using ensemble-based adversarial training for hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Md. Shihab Uddin Riad. 2025. Syntaxmind at BLP-2025 Task 1: Leveraging attention fusion of cnn and gru for hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Nauros Romim, Mosahed Ahmed, Md Saiful Islam, Arnab Sen Sharma, Hriteshwar Talukder, and Mohammad Ruhul Amin. 2022. BD-SHS: A benchmark dataset for learning to detect online Bangla hate speech in different social contexts. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5153–5162, Marseille, France. European Language Resources Association.

Nauros Romim, Mosahed Ahmed, Hriteshwar Talukder, and Md Saiful Islam. 2021. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020*, pages 457–468. Springer.

Sourav Saha, K M Nafi Asib, and Mohammed Moshiul Hoque. 2025. Retriv at BLP-2025 Task 1:a transformer ensemble and multi-task learning approach for bangla hate speech identification. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Mohammed Samir, Anisa Meem, and Faisal Abir. 2025. Team_nsu_strugglers at BLP-2025 Task 1: Multilevel approach to detect hateful speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Happy Khairunnisa Sariyanto, Diclehan Ulucan, Oguzhan Ulucan, and Marc Ebner. 2025. Towards explainable hate speech detection. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12883–12893.

S. Saumya, A. Kumar, and J. P. Singh. 2021a. Offensive language identification in dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, DravidianLangTech*, pages 36–45. Association for Computational Linguistics.

Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh. 2021b. Offensive language identification in Dravidian code mixed social media text. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 36–45, Kyiv. Association for Computational Linguistics.

Sad Yeamin Sayem and Sabira Rahman. 2025. Velora at BLP-2025 Task 1: Multi-method evaluation for hate speech classification in bangla text. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Salim Sazzed. 2021. Abusive content detection in transliterated bengali-english social media corpus. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 125–130, Online. Association for Computational Linguistics.

Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2021. Nlp-cuet@dravidianlangtech-eacl2021: Offensive language detection from multilingual code-mixed text using transformers. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, DravidianLangTech*, pages 255–261, Kyiv. Association for Computational Linguistics.

Deepawali Sharma, Tanusree Nath, Vedika Gupta, and Vivek Kumar Singh. 2025. Hate speech detection research in south asian languages: a survey of tasks, datasets and methods. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(3):1–44.

Shivam Sharma, Firoj Alam, Md. Shad Akhtar, Dimitar Dimitrov, Giovanni Da San Martino, Hamed Firooz, Alon Halevy, Fabrizio Silvestri, Preslav Nakov, and Tanmoy Chakraborty. 2022. Detecting and understanding harmful memes: A survey. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, IJCAI '22, pages 5597–5606, Vienna, Austria. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Intesar Tahmid, Rafid Ahmed, Md Mahir Jawad, Anam Borhan Uddin, Md Fahim, and Md Farhad Alam Bhuiyan. 2025. Pentaml @blp shared task 1: Linear probing of pre-trained transformer based models for bangla hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Hafsa Hoque Tripty, Laiba Tabassum, and Hasan Mesbaul Ali Taher. 2025a. Cuet_sntx_srfrs at BLP-2025 Task 1: Combining hierarchical classification and ensemble learning for bengali hate speech detection. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Zannatul Fardaush Tripty, Ibnul Mohammad Adib, Md. Tanjib Hossain, Nafiz Fahad, and Md. Kishor Morol. 2025b. Belite at BLP-2025 Task 1: Leveraging ensemble for multi-task hate speech detection in bangla. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Joseph B Walther. 2022. Social media and online hate. *Current Opinion in Psychology*, 45:101298.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Zeerak 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, San Diego, California. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yongjin Yang, Joonkee Kim, Yujin Kim, Namgyu Ho, James Thorne, and Se-Young Yun. 2023. HARE: Explainable hate speech detection with step-by-step reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5490–5505.

Samin Yasir. 2025. Heisenberg at BLP-2025 Task 1: Hate language detection from bangla comments on social media. In *Proceedings of the 2nd Workshop on Bangla Language Processing (BLP 2025)*, Mumbai, India. Association for Computational Linguistics.

Anwar Hossain Zahid, Monoshi Kumar Roy, and Swarna Das. 2025. Evaluation of hate speech detection using large language models and geographical contextualization. *arXiv preprint arXiv:2502.19612*.

A Leaderboard

We report the official leaderboard results for subtask 1B and 1C in Tables 4 and 5.

R.	Team Name	Dev P.	Eval. P.
1	TeamHateMate (Hasan and Mahim, 2025)	0.7536	0.7356
2	Code_Gen (Islam et al., 2025)	0.7532	0.7335
3	Gradient Masters (Rahman et al., 2025b)	0.7496	0.7328
4	Ecstasy (Hasan et al., 2025a)	0.7611	0.7317
5	SyntaxMind (Riad, 2025)	✗	0.7317
6	zannatul_007	0.7508	0.7315
7	abid_al_hossain	0.7448	0.7286
8	Catalyst (Hasan and Hasan, 2025)	0.7456	0.7279
9	BElite (Tripty et al., 2025b)	0.7560	0.7275
10	Retriv (Saha et al., 2025)	0.7500	0.7269
11	PentaML (Tahmid et al., 2025)	✗	0.7256
12	HateNetBN (Anam and Mazumder, 2025)	0.7249	0.7254
13	CUET-NLP_Zenith (Hossan et al., 2025)	0.7333	0.7213
14	adriti12	✗	0.7125
15	CoU-CU-DSG (Alam et al., 2025)	✗	0.7114
16	Computational StoryLab (Prama et al., 2025)	0.7492	0.7095
17	NSU_MILab (Rahman et al., 2025a)	0.7504	0.6981
18	PerceptionLab (Fahim and Khan, 2025)	✗	0.6979
19	pritampal98	0.7337	0.6974
20	Bahash-AI (Laskar and Paul, 2025)	✗	0.6954
21	CUET_Sntx_Srfrs (Tripty et al., 2025a)	✗	0.6817
22	Team_NSU_Strugglers (Samir et al., 2025)	0.6803	0.6760
23	Baseline (Majority)	0.6083	0.5974
24	lamiaa	✗	0.2848
25	Baseline (Random)	0.2118	0.2043
–	manik	0.7393	✗
–	no_name	0.7333	✗
–	Tensoryus	0.7277	✗
–	nur_163	0.7257	✗
–	rabeya_akter	0.7074	✗
–	unknown333	0.6361	✗
–	noob73	0.2647	✗
–	nuralflow	0.0565	✗

Table 4: Official ranking of the subtask 1B on the test set. – only participated in the Development Phase. ✗ indicates the team has not submitted the system in the respective phase. R.: Rank, Dev P.: Development Phase, Eval. P.: Evaluation Phase.

R.	Team Name	Dev P.	Eval. P.
1	TeamHateMate (Hasan and Mahim, 2025)	0.7554	0.7392
2	CUET-NLP_Zenith	0.7520	0.7378
3	Code_Gen (Islam et al., 2025)	0.7558	0.7361
4	Ecstasy (Hasan et al., 2025a)	0.7505	0.7332
5	BElite (Tripty et al., 2025b)	0.7561	0.7312
6	Gradient Masters (Rahman et al., 2025b)	0.7452	0.7310
7	Retriv (Saha et al., 2025)	0.7512	0.7262
8	abid_al_hossain	0.7404	0.7250
9	nur_163	0.7459	0.7241
10	Catalyst (Hasan and Hasan, 2025)	0.7459	0.7240
11	Computational StoryLab (Prama et al., 2025)	✗	0.7233
12	zannatul_007	0.7436	0.7181
13	PentaML (Tahmid et al., 2025)	0.7229	0.7159
14	pritampal98	0.7269	0.7153
15	abir_bot69	✗	0.7129
16	Team_NSU_Strugglers (Samir et al., 2025)	✗	0.7129
17	Bahash-AI (Laskar and Paul, 2025)	✗	0.6969
18	CUET_Sntx_Srfrs (Tripty et al., 2025a)	✗	0.6842
19	aacontest	✗	0.6730
20	Baseline (Majority)	0.6222	0.6072
21	adriti12	0.4141	0.3898
22	Baseline (Random)	0.2300	0.2304
–	foysal_ahmed	0.6939	✗
–	aacontest	0.6838	✗
–	unknown333	0.5669	✗
–	n00b	0.5464	✗

Table 5: Official ranking of the subtask 1C on the test set. – only participated in the Development Phase. ✗ indicates the team has not submitted the system in the respective phase. R.: Rank, Dev P.: Development Phase, Eval. P.: Evaluation Phase.