Dll5143A@LT-EDI 2025: Bias-Aware Detection of Racial Hoaxes in Code-Mixed Social Media Data (BaCoHoax)

Ashok Yadav and Vrijendra Singh
Indian Institute of Information Technology Allahabad, Prayagraj, 211015, India
{rsi2021002,vrij}@iiita.ac.in

Abstract

The proliferation of racial hoaxes that associate individuals or groups with fabricated crimes or incidents presents unique challenges in multilingual social media contexts. This paper introduces BaCoHoax, a novel framework for detecting race-based misinformation in code-mixed content. We address this problem by participating in the "Shared Task Detecting Racial Hoaxes in Code-Mixed Hindi-English Social Media Data: LT-EDI@LDK 2025." BaCoHoax is a bias-aware detection system built on a DeBERTa-based architecture, enhanced with disentangled attention mechanisms, a dynamic bias discovery module that adapts to emerging narrative patterns, and an adaptive contrastive learning objective. We evaluated BaCoHoax on the HoaxMixPlus corpus, a collection of 5,105 YouTube comments annotated for racial hoaxes, achieved a macro F1 score of 0.67, and secured 7th place among participating teams in the shared task. Our findings contribute to the growing field of multilingual misinformation detection and highlight the importance of culturally informed approaches to identifying harmful content in linguistically diverse online spaces.

1 Introduction

The proliferation of social media platforms has transformed how information spreads across diverse linguistic communities, creating both opportunities and challenges for marginalized populations worldwide ((Gowen et al., 2012); (Yates et al., 2017)). During recent global events, including the COVID-19 pandemic, online platforms have become critical spaces for emotional expression and support seeking (Wang and Jurgens, 2018). For vulnerable communities such as women in STEM, LGBTIQ individuals, racial minorities, and people with disabilities. These digital spaces significantly influence self-perception and societal integration (Chung, 2014); (Altszyler et al., 2018); (Tortoreto et al., 2019). While numerous studies have focused on detecting negative content through hate speech recognition (Schmidt and Wiegand, 2017), offensive language identification (Yadav et al., 2024), and implicit hate detection (Yadav and Singh, 2024), these approaches often fail to address underlying biases and may inadvertently discriminate against minority groups. The spread of misinformation targeting specific communities presents a particularly concerning challenge, especially in multilingual contexts where traditional detection systems struggle to operate effectively. This is especially problematic in code-mixed environments where individuals switch between multiple languages within single utterances (Chakravarthi, 2020), creating linguistic patterns that evade conventional detection methods. While research has expanded to include languages beyond English, including Arabic, German, Hindi, and Italian, these studies primarily examine monolingual corpora, overlooking the complexity of code-switched communication prevalent in diverse linguistic regions like South Asia.

The key contributions of our work include: (1) a DeBERTa-based architecture enhanced with disentangled attention mechanisms specifically optimized for code-mixed content; (2) a Dynamic Bias Discovery component that continuously identifies potentially biased terms throughout the training process, adapting to emerging narrative patterns; and (3) an adaptive contrastive learning approach that enhances the model's ability to distinguish subtle patterns in misinformation. Through comprehensive evaluation and error analysis, we demonstrate both the effectiveness of our approach. Code: [1]

[1] Code: https://github.com/ashokiiita/LDK_2025

Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 31–38
September 9, 2025 ©2025 Association for Computational Linguistics

2 Task and Dataset Description

The LT-EDI@LDK 2025 Shared Task focused on the detection of racial hoaxes in code-mixed Hindi-English content, addressing the critical challenge of misinformation that targets specific social or ethnic communities (Chakravarthi et al., 2025). The primary objective of the shared task was to develop computational systems capable of automatically identifying such content in low-resource, multilingual contexts. Participating teams were required to build binary classification systems that could distinguish between regular content (non-hoax) and racial based misinformation (hoax) in code-mixed text, where speakers switch between Hindi and English within the same utterance. The task evaluation employed macro-averaged F1 score as the primary metric, which equally weights performance on both the majority (non-hoax) and minority (hoax) classes, thereby encouraging systems to effectively identify the less frequent but more harmful racial hoax content.

The task organizers provided the HoaxMixPlus, a corpus comprising 5,105 YouTube comment annotated for racial hoaxes, as shown in Table 1. The HoaxMixPlus dataset is distributed across train (3,060), validation (1,021), and test (1,021) (Chakravarthi, 2020).

Table 1: Statistics of the dataset used in the LT-EDI@LDK 2025 Shared Task

Dataset	Total Samples	Non-Hoax (0)	Hoax (1)
Train	3060	2319	741
Validation	1021	774	247
Test	1021	774	247

3 Proposed Methodology

In this section, we present BaCoHoax, a novel framework for detecting racial hoaxes in code-mixed content. Our approach extends the capabilities of pretrained language models by integrating bias-aware components specifically designed to identify and leverage linguistic patterns associated with hoaxes. The framework consists of four main components: (1) a DeBERTa-based encoder, (2) a dynamic bias discovery mechanism, (3) disentangled bias-aware attention, and (4) a contrastive learning objective.

We formulate the task as a supervised classification problem. Given a text input $X = \{x_1, x_2, ..., x_n\}$ consisting of n tokens, our goal is to predict whether it contains a racial hoax, represented as a binary label $y \in \{0, 1\}$, where 1 indicates the presence of Hoax.

3.1 Base Architecture: DeBERTa-based encoder

Our model employs DeBERTa-v3 (He et al., 2021) as the backbone encoder due to its enhanced disentangled attention mechanism, which effectively separates content and positional information. This property is particularly valuable for our task, as it allows the model to better capture subtle contextual relationships in code-mixed content where sensitive terms may appear in varying positions and contexts. The encoder maps each input token to a contextualized representation, producing a sequence of hidden states $H = \{h_1, h_2, ..., h_n\}$. To capture richer semantic information, we employ a feature fusion approach that combines information from multiple layers using equation 1:

$$F = \text{LayerNorm}(W_f[H^L \oplus H^{L-1} \oplus H^{L-2}] + b_f) \quad (1)$$

where H^L, H^{L-1}, and H^{L-2} are the hidden states from the last three layers of DeBERTa, $\oplus$ denotes concatenation, and W_f and b_f are learnable parameters. This multi-layer fusion allows the model to leverage both high-level abstract features and lower-level syntactic information.

3.2 Dynamic Bias Discovery

In our approach we have used the DBD component, which continuously identifies potentially biased terms throughout the training process. Unlike static approaches that rely on predefined lexicons, DBD learns to recognize bias patterns directly from the data.

i. **Identity Term Identification:** We initialize the system with a seed list of common identity terms (e.g., religious, caste, and regional identifiers) in Hindi-English code-mixed text. For each input text, DBD identifies tokens that match predefined identity patterns.

ii. **Class Distribution Tracking:** For each identified term, DBD maintains a distribution of occurrences across the target classes (misinformation vs. factual), enabling the calculation of bias scores using equation 2.

$$\text{BiasScore}(t) = \frac{\text{count}(t, y = 1)}{\text{count}(t, y = 0) + \text{count}(t, y = 1)} \quad (2)$$

iii. **Contextual Embedding:** For each bias term, DBD stores and updates representative embeddings by averaging the contextualized representations of the term across all its occurrences.

iv. **Threshold-based Selection:** Terms with strong class associations (bias scores significantly above or below 0.5) and sufficient occurrence count are added to the bias term set.

This component allows our model to adaptively discover new bias terms without manual annotation, making it robust to evolving language patterns and novel bias expressions.

3.3 Disentangled Bias Detection and Attention

To effectively leverage the discovered bias terms, we implement two specialized components:

3.3.1 Disentangled Bias Detector

The DBD identifies potential bias signals in the input text by comparing token embeddings with the embeddings of known bias terms. For each token in the input sequence, DBD computes a bias probability by identifying tokens that match known bias terms, then extracting a context window around each identified token. It subsequently applies disentangled attention to evaluate the contextual usage of each term, and finally computes a bias probability for the token along with its surrounding context. This results in a bias mask $M_{bias} \in \mathbb{R}^n$ that highlights tokens likely to contribute to biased or misleading content.

3.3.2 Disentangled Bias Attention

The Disentangled Bias Attention (DBA) module extends DeBERTa's disentangled attention mechanism by incorporating bias awareness:

$$A_{i,j} = \frac{(W_Q h_i)^T (W_K h_j)}{\sqrt{d}} + \alpha \cdot M_{bias}[j] \quad (3)$$

where W_Q and W_K are query and key projection matrices, d is the dimension of the projection, and α is a learnable parameter that controls the influence of the bias mask. This attention mechanism amplifies focus on tokens identified as potentially biased, allowing the model to better analyze their contextual usage.

The bias-aware context representation is computed using equation 4:

$$C_{bias} = \text{softmax}(A) \cdot (W_V H) \quad (4)$$

This representation is then combined with the standard sentence representation to form a comprehensive feature vector.

3.4 Classification and Learning Objectives

Our model employs a multi-objective learning approach that combines classification and contrastive learning:

The classification head takes the concatenated representation of the [CLS] token embedding and the bias-guided vector:

$$Z = [h_{CLS} \oplus v_{bias}] \quad (5)$$

This is passed through a multi-layer classifier to produce class logits:

$$\hat{y} = \text{softmax}(W_c \text{MLP}(Z) + b_c) \quad (6)$$

where MLP is a multi-layer perceptron with layer normalization and GELU activation functions.

To enhance the model's ability to distinguish subtle patterns in hoax content, we implemented an adaptive contrastive learning objective. For each input sample, we create an augmented version using DeBERTa-specific augmentation techniques such as Entity emphasis (duplicating entity mentions), Word deletion (removing non-essential words) and Synonym replacement (substituting words with similar meanings).

These enhancements preserve semantic content while creating variations that help the model learn robust representations. The contrastive loss is computed using an adaptive temperature scaling approach using equation 7.

$$\mathcal{L}_{cont} = -\frac{1}{|P|} \sum_{(i,j)\in P} \log \frac{\exp(z_i \cdot z_j/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{[k\neq i]} \exp(z_i \cdot z_k/\tau)} \quad (7)$$

where P is the set of positive pairs, z_i and z_j are normalized projection embeddings, and τ is an adaptive temperature parameter adjusted based on batch similarity distribution. The final loss combines classification and contrastive objectives using equation 8.

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{cont} \quad (8)$$

where λ is dynamically adjusted during training based on validation performance, allowing the model to find an optimal balance between the two objectives.

4 Experimental Settings and Result Analysis

We implement our model using PyTorch and the Transformers library. For the DeBERTa encoder, we use the `microsoft/deberta-v3-base` variant with 12 layers and a hidden size of 768. We set the maximum sequence length to 128 tokens and use a batch size of 8 for training. The AdamW optimizer is configured with a base learning rate of 2×10^{-6} and a maximum learning rate of 1×10^{-5}, with a weight decay of 0.01. We train the model for 10 epochs with early stopping based on the F1 validation score. For the bias discovery component, we use a bias threshold of 0.65 and a minimum occurrence count of 5. The contrastive learning weight is initialized at 0.15 and dynamically adjusted during training within the range [0.05, 0.3]. All experiments were performed using a NVIDIA A30 GPU.

The performance of our BaCoHoax framework was evaluated in the context of the shared task, where systems were primarily ranked based on their macro-averaged F1-scores. Table 2 summarizes the official results and standings of all participating teams.[2]

Table 2: Official Ranking of Teams in the HoaxMixPlus Shared Task

Team Name	Macro F1 Score	Rank
CUET's_White	0.75	1
Hope_for_best	0.72	2
KCRL	0.71	3
HoaxTerminator	0.70	4
Hinterwelt	0.69	5
Belo Abhigyan	0.68	6
KEC-Elite-Analytics	0.68	6
DlI5143A	**0.67**	**7**

Our BaCoHoax model achieved a macro F1 score of 0.67, securing 7th position among all participating teams. This performance demonstrates the effectiveness of our approach in addressing the challenging task of detecting racial hoaxes in code-mixed content. The top-performing system (CUET's_White) achieved a macro F1 score of 0.75, The relatively tight clustering of scores among the top 8 teams (ranging from 0.75 to 0.63) suggests that detecting racial hoaxes in code-mixed Hindi-English content remains challenging, with multiple approaches achieving similar performance levels. The detailed results are discussed in the Appendix A.1

5 Conclusion and Future Work

In this paper, we introduced BaCoHoax, a novel framework for detecting race based Hoax content in code-mixed content that leverages disentangled attention mechanisms and bias-aware representations. Our approach incorporates cultural and linguistic nuances through a dynamic bias discovery mechanism and contextual understanding of bias terms in multilingual settings. We achieved a competitive macro F1 score of 0.67 and securing 7th place among participating teams in the shared task. The pre-defined bias term approach creates problematic overgeneralization and identity-bias conflation, where the model cannot effectively distinguish between merely mentioning an identity group and expressing bias toward that group.The imapct of pre-deifned bias terms are discussed in detail A.4. These findings suggest the need for context-aware and culturally grounded bias detection methods, along with improved discourse analysis and refined lexicon strategies. Future models should also consider multi-stage architectures to better handle the complexity of code-mixed and nuanced biased content.

6 Error Analysis

We conducted detailed error analysis which provides insights of specific areas where our model faced difficulties, particularly in detecting subtle forms of racial bias expressed through cultural references, rhetorical devices, and implicit language challenges. The detailed error analysis is dicussed in Appendix A.2. results validate the effectiveness of our BaCoHoax framework while highlighting opportunities for further refinement in future iterations.

References

Edgar Altszyler, Ariel J Berenstein, David N Milne, Rafael A Calvo, and Diego Fernandez Slezak. 2018. Using contextual information for automatic triage of posts in a peer-support forum. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 57–68.

[2] Official leaderboard: https://codalab.lisn.upsaclay.fr/competitions/21885#learn_the_details-evaluation

Bharathi Raja Chakravarthi. 2020. Hopeedi: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53.

Bharathi Raja Chakravarthi, Prasanna Kumar Kumaresan, Shanu Dhawale, Saranya Rajiakodi, Sajeetha Thavareesan, Subalalitha Chinnaudayar Navaneethakrishnan, and Durairaj Thenmozhi. 2025. Overview of the shared task on detecting racial hoaxes in code-mixed hindi-english social media data. In *Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity and Inclusion*, Italy. Fifth Conference on Language, Data and Knowledge (LDK2025).

Jae Eun Chung. 2014. Social networking in online support groups for health: how online social networking benefits patients. *Journal of health communication*, 19(6):639–659.

Kris Gowen, Matthew Deschaine, Darcy Gruttadara, and Dana Markey. 2012. Young adults with mental health conditions and social networking websites: seeking tools to build community. *Psychiatric rehabilitation journal*, 35(3):245.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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

Giuliano Tortoreto, Evgeny A Stepanov, Alessandra Cervone, Mateusz Dubiel, and Giuseppe Riccardi. 2019. Affective behaviour analysis of on-line user interactions: Are on-line support groups more therapeutic than twitter? *arXiv preprint arXiv:1911.01371*.

Zijian Wang and David Jurgens. 2018. It's going to be okay: Measuring access to support in online communities. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 33–45.

Ashok Yadav, Farrukh Aslam Khan, and Vrijendra Singh. 2024. A multi-architecture approach for offensive language identification combining classical natural language processing and bert-variant models. *Applied Sciences*, 14(23):11206.

Ashok Yadav and Vrijendra Singh. 2024. Hatefusion: Harnessing attention-based techniques for enhanced filtering and detection of implicit hate speech. *IEEE Transactions on Computational Social Systems*.

Andrew Yates, Arman Cohan, and Nazli Goharian. 2017. Depression and self-harm risk assessment in online forums. *arXiv preprint arXiv:1709.01848*.

A Appendix

A.1 Appendix A: Result Analysis

Table 3 presents the detailed performance metrics of the BaCoHoax model on the training set. The model achieves an overall accuracy of 86%, indicating that it correctly classifies approximately four out of five instances in the training data. However, this aggregate metric masks significant disparities in the model's ability to detect different classes of content.

For non-racial content (class 0), the model demonstrates strong performance with a precision of 0.88 and recall of 0.93, resulting in an F1-score of 0.91. This indicates that when the model predicts content as non-racial, it is correct 88% of the time, and it successfully identifies 93% of all non-racial content in the dataset. These metrics suggest that the model has developed a robust understanding of non-racial content patterns. In contrast, the model's performance on racial content (class 1) is considerably less stronger. With a precision of 0.75 and recall of only 0.61, the resulting F1-score is a modest 0.67. The macro-average metrics (precision: 0.81, recall: 0.77, F1-score: 0.79) provide a class-balanced view of performance and highlight the significant room for improvement in identifying racial content. The weighted averages (precision: 0.76, recall: 0.79, F1-score: 0.77) appear somewhat higher due to the dominance of non-racial samples in the dataset and the model's stronger performance on this majority class. These training set metrics highlight a core challenge in the bias detection task. While the model performs well in identifying non-racial content, it struggles to detect racial bias, particularly when it is expressed in subtle or implicit forms.

Table 3: Training Set Performance Metrics of the BaCoHoax Model

Class	Precision	Recall	F1-Score
Non-Racial (0)	0.88	0.93	0.91
Racial (1)	0.75	0.61	0.67
Accuracy		0.86	
Macro Average	0.81	0.77	0.79
Weighted Average	0.85	0.86	0.85

Table 4 presents the performance metrics of our BaCoHoax model on the validation set. The model overall accuracy of 75%, indicating good performance on unseen data. For non-racial content (class 0), the model demonstrates robust performance with a precision of 0.83 and an exceptionally high recall of 0.85, resulting in an F1-score of 0.84. This indicates that the model effectively recognizes 85% of all non-racial content in the validation set, and when it classifies content as non-racial, it is correct 83% of the time. These results confirm that the model has developed a strong capacity to identify non-racial content patterns that generalize well to unseen data.

However, the model's performance on racial content (class 1) shows significant weaknesses. With a precision of 0.49 and a very low recall of 0.45, the resulting F1-score is only 0.47. These metrics reveal a critical limitation in the model's ability to detect racially-charged content. The macro-average F1-score of 0.65 provides a class-balanced performance metric that emphasizes the substantial performance gap between classes. The weighted average F1-score of 0.75 appears higher primarily due to the class imbalance in the dataset and the model's stronger performance on the majority class.

The validation results reveal a persistent challenge in racial content detection, while the model generalizes well for non-racial content. The lower recall on racial content compared to the training set suggests that the model may be encountering forms of racial bias in the validation set that differ from those in the training data, highlighting the difficulty of capturing the diverse and evolving expressions of racial bias in code-mixed text.

Table 4: Validation Set Performance Metrics of the BaCoHoax Model

Class	Precision	Recall	F1-Score
Non-racial (0)	0.83	0.85	0.84
Racial (1)	0.49	0.45	0.47
Accuracy		0.75	
Macro Average	0.66	0.65	0.65
Weighted Average	0.75	0.75	0.75

Table 5 presents the final performance metrics of our BaCoHoax model on the held-out test set. The model achieves an overall accuracy of 76%. For non-racial content (class 0), the model exhibits strong performance with a precision of 0.85 and recall of 0.84, resulting an F1-score of 0.84. These metrics show that the model effectively identifies 84% of all non-racial content in the test set, and when it classifies content as non-racial, it is correct 85% of the time.

However, the model's performance on racial content (class 1) continues to show substantial limitations. With a precision of 0.51 and a recall of 0.53, the resulting F1-score is 0.52. These figures indicate that the model fails to identify 47% of racial content in the test data, and when it does predict content as racial, it is incorrect 49% of the time. The macro-average F1-score of 0.67 and weighted average F1-score of 0.77 are consistent with those observed in the training set, confirming the model's stable performance pattern across dataset. The significant gap between non-racial and racial content detection capabilities remains a persistent characteristic of the model. The test results confirm that while the BaCoHoax approach provides a solid foundation for racial bias detection, significant improvements are still needed to address the challenge of identifying subtle and implicit forms of racial bias.

Table 5: Test Set Performance Metrics of the BaCoHoax Model

Class	Precision	Recall	F1-Score
Non-racial (0)	0.85	0.84	0.84
Racial (1)	0.51	0.53	0.52
Accuracy		0.76	
Macro Average	0.68	0.68	0.67
Weighted Average	0.77	0.76	0.77

A.2 Appendix

B: Error Analysis

A.2.1 Analysis of False Positive Errors

We analyzed instances where our model incorrectly classified non-racial content as containing racial bias. Table 6 presents a selection of these false positive cases.

Our analysis of false positive cases reveals consistent patterns in the model's tendency to overclassify content as racially biased. These patterns underscore key limitations in the current approach to bias detection, particularly in the context of code-mixed text. One prominent issue is the model's frequent misinterpretation of general profanity and aggressive language as signals of racial bias. For instance, in example RH_10, the text criticizes the news media using terms like "chor gadar atankwadi" (thief, traitor, terrorist), yet it does not target any specific religious or ethnic group. Similarly, RH_30 includes profane insults such as "shale suar ki olad" but focuses on disrupting societal peace rather than targeting protected communities.

Table 6: Examples of False Positive Errors in the BaCoHoax Model

ID	Text Content	Predicted	True
RH_10	tum sare news walo chor gadar atankwadi ho jo sirf moh se zaher ugalna janti ho	1	0
RH_18	kiyo bat karte ho musalman ko pakistan jane ki esa hi hoga sarkar kadam nahi utaygi to majburan bolna padta hai kiyo army nahi bejte brma mei bacho ko kata ja rha hai ladies ki ijat loti ja rhi hai fast bola us par dyan diya nahi chutya kahi k ai hai bahsan dene k liye	1	0
RH_30	shale suar ki olad teri okat kya h ham desh me shanti chahte h barna sudhar jao salo	1	0
RH_35	congres party ke gadar desh drohi hai aise admi ko goli mardo madarchod sale ne bhrashtachar kiya hai...	1	0
RH_38	bihar in purani kaduai yado se age nikal chuka hai apne 10, 20 like coment ke like ye sab na kare to apne bihar ke like this hai	1	0
RH_51	modi ki hrkatu se yah sabit hua ki mohamad ali jinah thek tha aur abdul kalam azad ghalat hindustan ki musalmanon ka mujrim abdul kalam azad hai aur koi nai jis ne pakistan ki bajhe hindustan ka sathiya	1	0
RH_68	ak muslim hote hue me kh raha hu ke es truke ki jumla baji krne valo ko desh ka savidhan bade se bdi saja de...	1	0
RH_109	islam kaha se peda huva alah khush hoke peda kiya jatiwadi one jat ke nam logo ka shoshan atyachar...	1	0
RH_119	mujhe hindu musalman nahinsan aur insaniyat kahatam hote dikh rahi haialah har bande ko nek hidayat...	1	0
RH_129	to hinduo bijli pani fri ke liye kejriwal ko vote b tumne hi kiya tha ab jutey khao	1	0

These examples suggest that the model exhibits a problematic bias by linking profanity with race. A second pattern involves the misclassification of political criticism as racial bias. In RH_35, the model flags a harsh critique of the Congress party using the phrase "gadar desh drohi" (traitor, anti-national) and referencing corruption, even though there is no mention of race or religion. This indicates that the model may be over-associating political discourse with bias, possibly due to correlations in the training data where political references frequently co-occurred with biased content.

The model also struggles to distinguish between neutral or positive mentions of identity groups and actual bias. In RH_68, a self-identified Muslim user condemns divisive rhetoric and advocates for peace and harmony. Similarly, RH_119 explicitly states, "mujhe Hindu Musalman nahin, insaan aur insaniyat" (I don't see Hindu/Muslim, but humanity), yet the model still classifies the text as biased. These misclassifications suggest that the model relies heavily on the presence of identity terms without fully understanding their contextual meaning. Another common source of error involves historical or factual references to sensitive topics. In RH_51, the speaker discusses historical figures like Jinnah and Azad, while RH_109 explores the origins of Islam and caste-based oppression. Neither example promotes contemporary bias, yet both are flagged by the model.

We also observed that texts containing contentious arguments involving communities but lacking explicit bias which are often misclassified. RH_38 critiques the resurfacing of past divisions in Bihar, and RH_129 questions the political choices of Hindu voters without expressing religious animosity. These cases reflect a broader issue: the Disentangled Bias Detector tends to flag content containing identity-related keywords (e.g., “Hindu,” “Muslim,” “BJP,” “Congress”) regardless of context, especially when such terms frequently co-occurred with biased content during training. Similarly, references to historical conflicts are often treated as biased, even when discussed factually. These findings support our hypothesis that the use of predefined bias terms during model initialization may contribute to overgeneralization and the conflation of identity mention with bias. Limitations in contextual understanding are particularly evident in cases like RH_68 and RH_119, where the model detects religious terms but fails to recognize the underlying pro-harmony messages.

A.3 False Negative Analysis

Following our analysis of false positive cases, we conducted analysis of false negatives samples to further assess the limitations of the BaCoHoax model. Specifically, we analyzed instances where racially biased hoax content was incorrectly classified as non-biased. Table 7 presents representative examples of such misclassifications, illustrating the model's difficulty in recognizing subtle or contextually embedded racial bias.

Our analysis of these false negative cases reveals several consistent patterns in the model's inability to detect racial content. In example RH_07,

Table 7: Examples of False Negative Errors in the BaCoHoax Model

ID	Text Content	Predicted	True
RH_07	behan sambhal ke rahana in ka yeisa hai ki jab ap inki bat karoge ye thik hai par inki ashliyet batoge to tunhehi khatra paida karenge ye kisan nahi khalisthani hai...	0	1
RH_26	hamara desh ke khate mein police hamara hinduon ka tax do per hi atyachar hote hain yah kaun si bat ho gai ise banaoge hindu rashtra ke	0	1
RH_31	hasi ati h tere knowledge pe puri duniya me including india sab se bda inflict muslims me hi h	0	1
RH_40	ye khabar sunkar sch bta rha hu hath m talwar ho aur fir s unko katu jo in garibo ki aurat ko bhi nhi baksa aur inke jatiwad ko inki gand m dal deta fir dhekte h ki kon asli mard h	0	1
RH_48	teri soch kitni nech hai tere ma bap ki trha surnames kisi ke bap ki bapoti nhi hai jiska jo mn chahega wo wo likhega verma likhe ya harijan uski marzi aur shadi ke samay sb apni caste batate hai koi kuch nhi chupata lgta hai teri mumy ya pita ne aise hi shadi ki hogi bhag kr	0	1
RH_50	bhai ye musalman ki fake id hai delhi election mai in katwo ko pta chal jayega	0	1

the model fails to identify the implicit bias where protestors are characterized as "khalisthani" (a politically charged term) rather than legitimate farmers. Similarly, in RH_50, the model misses the derogatory reference to Muslims using the term "katwo," which is highly offensive in the given context. Many of these examples demonstrate the model's difficulty in recognizing coded language and cultural references that require deep understanding of the sociopolitical context.

For instance, RH_26 contains subtle implications about tax contributions and religious identity that require contextual understanding beyond simple term matching. The model also struggles with complex cases like RH_31 makes broad negative generalizations about Muslims that the model fails to detect, likely due to the absence of explicit slurs or recognized bias terms. Additionally, examples like RH_40 and RH_48 demonstrate the model's inability to identify violent rhetoric (references to weapons) and caste-based insults when they're embedded within colloquial expressions or aggressive language patterns common in social media discourse. These patterns highlight the need for more sophisticated approaches to bias detection that can capture subtle linguistic cues, cultural references, and implicit expressions of prejudice in code-mixed text. The evolving and evasive nature of biased language makes detection difficult.

A.4 Impact of Pre-defined Bias Terms

A critical factor in both false positive and false negative errors can be traced to the model's initialization with predefined bias terms. The BaCoHoax model was seeded with an initial set of bias terms including identity markers (“hindu”, “muslim”, “islam”, “mandir”, “masjid”, “brahmin”, “dalit”, “sc”, “st”, “obc”), names associated with specific communities (“singh”, “khan”, “sharma”, “ali”, “kumar”), politically charged terms (“jihad”, “bhakt”, “sanghi”, “liber”), and political party references (“congress”, “bjp”, “modi”, “rahul”). This initial seeding creates several problematic effects that significantly impact model performance. The approach leads to overgeneralization, as the model is primed to flag any content containing these broad identity and political terms regardless of context. It also results in identity-bias conflation, where the model cannot effectively distinguish between merely mentioning an identity group and expressing bias toward that group, creating a fundamental “guilt by association” problem. Additionally, the inclusion of political terms as bias markers creates a problematic political-communal confusion, explaining many false positives in content that offers political criticism without communal bias. The model's bias detection mechanism demonstrates context-free processing, assuming these terms are inherently biased rather than recognizing their meaning is highly context-dependent. Furthermore, the Dynamic Bias Discovery component creates a self-reinforcing bias loop where neutral mentions may be incorrectly classified as biased, strengthening problematic associations over time. For false negatives, this approach fails to capture bias expressed through more subtle means without explicitly using the predefined terms, while for false positives, it overreacts to any content containing these terms regardless of context or intent.