Can native language samples improve code-mixed hate detection?: A case study for Hindi-English code-mixed hate detection

Anonymous ACL submission

Abstract

Hate detection has long been a challenging task for the NLP community. The task becomes complex in a code-mixed environment because the models must understand the context and the hate expressed through language alteration. Compared to the monolingual setup, we see very less work on code-mixed hate as large-scale annotated hate corpora are unavailable to make the study. To overcome this bottleneck, we propose using native language hate samples. We hypothesise that in the era of multilingual language models (MLMs), hate in code-mixed settings can be detected by majorly relying on the native language samples. Even though the NLP literature reports the effectiveness of MLMs on hate detection in many cross-lingual settings, their extensive evaluation in a code-mixed scenario is yet to be done. This paper attempts to fill this gap through rigorous empirical experiments. We considered the Hindi-English code-mixed setup for our study, and some of the interesting observations we got are: (i) adding native hate samples in the code-mixed training set, even in small quantity, improved the performance of MLMs for code-mixed hate detection, (ii) MLMs trained with native samples alone observed to be detecting code-mixed hate to a large extent, (iii) The visualisation of attention scores revealed that, when native samples were included in training, MLMs could better focus on the hate emitting words in the code-mixed context, and (iv) finally, when hate is subjective or sarcastic, naively mixing native samples doesn't help much to detect code-mixed hate. We have shared the data and code repository to reproduce the reported results.

Keywords – Code-mixed hate detection, Cross-lingual learning, Native sample mixing.

1 Introduction:

The rising cases of online hate-speech (2023) and similar components (cyberbullying (2012), racism (2016), gender discrimination (2017), radicalisation (2013), religious hatred, abusive language detection (2018) etc.) distress the sanity and the civic nature of online discussions. To address this, the NLP community have been working long on characterisation, detection and mitigation of these components (2017; 2018; 2019; 2019; 2021; 2020; 2023; 2023; 2023). The intensity of their focus can be gauged by the fact that around 460+ peer-reviewed AI/ML papers were published between 2001 and 2021 (2023) on this topic, and exponentially rising since then. The community has proposed 70+ datasets spanning 20+ languages and different modalities (memes, texts, social media posts, images, videos etc.) (2023; 2023).

Even now, we have yet to eradicate the hate completely from online platforms. The main reason is that most of the studies were done for a few resource-rich languages (51% alone for English), whereas we have 100+ languages in the world, each having seven million+ speakers[1]. Further, a small fraction of these studies focused on code-mixed setups where the hate is uttered altering more than one language. It is common in multilingual environments like Europe, India and Latin America, where a significant portion of the population knows more than one language. Hate detection in code-mixed language is more complex compared to the monolingual environment. The models must understand the context and the hate expressed through multiple languages. One of the primary bottlenecks for the research on code-mixed hate detection is the unavailabil-

[1] http://www2.harpercollege.edu/mhealy/g101ilec/intro/clt/cltclt/top100.html

ity of large-scale training corpora. The same reason also holds for most of the low-resource languages.

The emergence of multilingual language models (MLMs) allowed us to address this issue via cross-lingual learning. It enables the models to learn task-specific knowledge from a dataset in one language and make predictions for samples in different languages. Past works demonstrated that MLMs, to a significant extent, could detect hate when train and test languages are different (2022). This is because (i) the vocabulary of MLMs covers many languages, and (ii) their embeddings encode the semantic and syntactic features seen across multiple languages. However, they face several drawbacks too. The NLP literature reported (2021; 2022) that cross-lingual learning fails when hate (i) expresses language-specific taboos and (ii) is specific to a community or culture. In the context of code-mixed hate detection, a thorough evaluation of MLMs and cross-lingual learning is lacking. This paper attempts to fill this gap by doing extensive empirical experiments. Particularly, we attempted to answer the following two research questions,

- **R1:** *Does training with additional native hate samples impact the code-mixed hate detection?*

- **R2:** *Can training with only native samples detect the hate in a code-mix scenario?*

All of our experiments were done for the Hindi-English code-mixed environment. This is because, only for this pair, we could find annotated code-mixed and respective native language hate corpora publicly available. Our contributions in this paper can be summarised as,

- We evaluated the impact of native language hate samples on code-mixed hate detection (**Exp 1:** Section 3.2) by comparing the performance across two training sets, (i) having only code-mixed hate samples and (ii) with additional samples from Hindi and English hate corpora. We experimented with two types of models: (i) statistical classifiers on top of the word n-grams and (ii) MLMs such as mBERT (2019), XLM (2019) and XLM-R (2019) with and without additional transformer layers. We reported the model performance when native samples were added, with an equal label distribution or with a label ratio the same as in the code-mixed hate corpora. Further, we also reported the performance variations of the MLMs when native samples were added in different amounts (**Exp 2:** section 3.3).

- We evaluated the performance of MLMs by training them only with native samples (**Exp 3:** Section 3.4) as well. We created three types of training sets: (i) with only Hindi samples, (ii) with only English samples and (ii) with Hindi and English samples together.

- We visualised the attention scores given by the MLMs and reported the change in scores on hate-emitting words after native samples were mixed with the original code-mixed training set (Figure 1).

- Finally, we manually inspected each test prediction and reported the cases for which MLMs (a) got better, (b) remained confused, and (c) performed worse after native samples were mixed in the code-mixed training set.

Some of the interesting results we got are,

- On combining the native samples, while the statistical models performed worse, many MLMs reported significant ($p < 0.05$) improvements (an increase of ∼**0.09** in $F1$ score)

- MLMs, when trained with only native samples, could identify hate in a code-mixed context nearly as good as when trained with additional code-mixed samples (∼**0.6** in $F1$ score for both). It implies that we can deploy the MLMs trained with native samples if an appropriate code-mixed corpus is unavailable. Further, we also observed that the native language dataset that shares maximum demographic or cultural overlap with the code-mixed hate corpus is more prominent in capturing the code-mixed hate.

- We observed that after native sample mixing, the MLMs gave high attention scores to the hate-emitting words when they appeared in the code-mixed context. Some of such cases were demonstrated in Figure 1.

- Finally, in the error analysis, we also reported cases where hate is sarcastic and subjective; it's still hard to identify them in the code-mixed setting by simply relying on native hate samples.

2 Dataset details:

We considered the publicly available Hindi-English code-mixed hate dataset (2018a), made up of social media posts. The authors retrieved 1,12,718 tweets based on a list of hashtags and keywords related to politics, public protests, riots, etc. Out of which, they manually filtered out 4575 code-mixed tweets. Expert annotators manually annotated the filtered tweets as "hate" or "non-hate". The label distribution is reported in Table 1.

Additionally, for native language samples, we considered publicly available Hindi (2022) and English [2] hate datasets in our experiments. We chose them as they are (i) relatively balanced, (ii) versatile, and (iii) widely used in past works. Their samples are manually annotated as "hate"/ "non-hate". The observed label distributions for both are reported in Table 1. The supplementary report presents some samples from all three considered datasets.

Dataset	Hate	Non-Hate
English-Hindi CM(2018a)	1661	2914
English[2]	2261	3591
Hindi(2022)	3338	1416

Table 1: Distribution of labels present in the considered code-mixed and monolingual hate corpora.

3 Experiments:

3.1 Experimental set-up:

We considered three generic architectures for our experiments. Note that our aim was not to propose any novel architecture; rather, we evaluated the behaviour of widely used statistical and MLM-based models when native samples were kept in the training sets. The three architectures we considered are,

1. Statistical classifiers SVM, Random Forest and Naive Bayes on top of word n-gram features. We considered word-level unigram, bigram and trigram features, as past work (2018a) reported them to perform best.

2. As a second approach, we fine-tuned MLMs mBERT (2019), XLM-R (2019) and XLM (2019). We stacked a linear layer on top of the [CLS] token embedding and fine-tuned their last two layers.

3. Lastly, we stacked four transformer layers followed by a linear layer on top of the considered MLMs (represented by mBERT$_{trans}$, XLM-R$_{trans}$ and XLM$_{trans}$ respectively). During fine-tuning, we froze all but the last two layers of the language model.

We collected the pre-trained weights for the MLMs from the Huggingface[3] transformer library. We used AdamW optimizer (2017), with a learning rate of 2×10^{-5}, and a scheduler with a learning rate decay of 0.9(gamma value) for training. We experimented with various (i) batch sizes, (ii) random seeds and (iii) early stopping strategies. Models were trained for a maximum of 25 epochs with an epoch patience of four.

3.2 Exp. 1- Impact of added native samples:

In the first experiment, we measured the impact of adding native samples on code-mixed hate detection. We created training(70%), validation(15%) and test(15%) splits of the code-mixed hate dataset (2018a) using stratified sampling. Additionally, we formulated two new training sets by including samples from the English and Hindi hate corpora. In the first set (**Train-1**), we added an equal amount of "hate" or "non-hate" samples from each monolingual dataset. In the second (**Train-2**), we kept label distribution the same as we observed in the training split of the code-mixed dataset. The label distribution present in the formulated training sets, in the initial code-mixed

[2] HASOC-2019: https://hasocfire.github.io/hasoc/2019/dataset.html

[3] https://huggingface.co/

training set, and in the validation and test sets are reported in table 2.

Partition	**Dataset**	**Hate**	**Non Hate**
Train-1	CM	1149	2062
	Hindi	1416	1416
	English	1416	1416
	Total	3981	4894
Train-2	CM	1149	2062
	Hindi	810	1416
	English	2000	3500
	Total	3959	6978
Validation	CM	249	438
Test	CM	249	437

Table 2: The distribution of labels present in the formulated training sets **Train-1** (equal ratio) and **Train-2** (code-mixed ratio). The initial code-mixed training set distribution is presented next to **CM** in Train-1 and Train-2. The numbers next to **Validation** and **Test** represent the distribution of labels in code-mixed validation and test splits, respectively.

3.3 Exp. 2- Impact of native samples added in different amounts:

Our second experiment quantified the impact of English and Hindi samples when they were added in different amounts to the code-mixed training set. For this purpose, we created two bags of training sets. In the first bag, training sets were created by incrementally adding two batches of native samples with the code-mixed training set. One of the batches had English samples, while the other had Hindi samples. Each of them had 200 randomly selected unique samples with an equal ratio of “hate” and “non-hate” samples taken from the respective monolingual hate corpora (similar to **Train-1** in the first experiment). In the second bag, we created similar training sets, except here, the label ratio in each batch was the same as in the original code-mixed training set (similar to **Train-2** in the first experiment).

3.4 Exp. 3- Relying only on native samples:

Our last experiment evaluated the performance of MLMs after they were trained with only native samples. The intuition behind this experiment was to check if the knowledge learned from the training of native samples was enough to detect hate in the code-mixed scenario. We considered three types of training sets: (i) with only Hindi samples, (ii) with only English samples, and (iii) with both English and Hindi samples. All of the training sets had an equal amount of “hate” and “non-hate” samples. We fine-tuned the hyper-parameters on the code-mixed validation set created as a part of the first experiment.

4 Results and discussion

4.1 Exp.-1 observations:

Here, we reported the results of our experiment measuring the impact of native hate samples on code-mixed hate detection. We compared the performance of different models in terms of accuracy (Acc), precision (Pre), recall (Rec) and F1-score ($F1$), and reported them in Table 3. The best scores for individual training scenarios were marked in bold. Since the label distributions in all training sets are unbalanced, we believe the $F1$ score best compares the performance. Following were our takeaways,

- On combining the native language samples, statistical models got confused and performed worse. On the contrary, MLMs significantly improved (an increase of 0.07 and 0.09 in best-reported $F1$ scores for Train-1 and Train-2, respectively). The MLMs for which the improvements were statistically significant (t-test, $p < 0.05$) are marked with an asterisk ($\ast$) next to their $F1$ scores.

- We observed minimal or no $F1$-score improvement when transformer layers were added to the MLMs. This indicates that fine-tuning the last layers of MLMs can serve the purpose to a large extent.

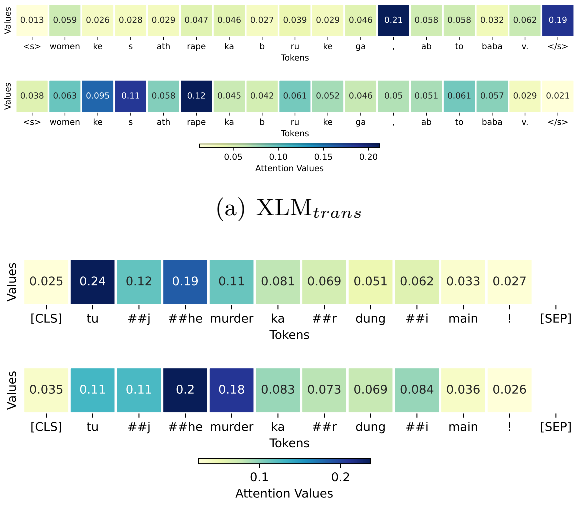
- We observed no clear performance difference in MLMs when native language samples were mixed in different label ratios (**Train-1** vs **Train-2**). It indicates that they are relatively robust towards label unbalancing.

- We investigated the attention scores given by MLMs after they were trained with or without the native samples. We observed that after native sample mixing, the MLMs gave high scores to the hate-emitting words when they appeared in

Models	Codemix				Train-1				Train-2			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
SVM	0.67	0.58	0.37	0.45 (±0.03)	0.65	0.51	0.39	0.44 (±0.01)	0.69	0.60	0.40	0.48 (±0.01)
RF	0.67	**0.77**	0.13	0.23 (±0.01)	0.67	**0.71**	0.16	0.26 (±0.00)	0.66	**0.79**	0.10	0.18 (±0.02)
Naive Bayes	0.63	0.49	0.53	0.51 (±0.02)	0.66	0.65	0.16	0.26 (±0.02)	0.67	0.62	0.25	0.36 (±0.01)
XLM	0.68	0.52	**0.56**	**0.52** (±0.01)	0.69	0.58	0.47	0.52 (±0.01)	0.67	0.53	0.63	0.58* (±0.01)
mBERT	0.61	0.44	0.55	0.49 (±0.01)	**0.71**	0.62	0.51	0.56* (±0.02)	0.68	0.57	0.55	0.56 (±0.00)
XLM-R	0.67	0.51	0.43	0.47 (±0.02)	0.67	0.54	0.61	0.58* (±0.01)	0.69	0.60	0.46	0.52 (±0.01)
XLM$_{trans}$	0.65	0.48	0.52	0.50 (±0.02)	0.69	0.56	0.63	0.59* (±0.01)	0.67	0.53	**0.65**	0.58* (±0.01)
mBERT$_{trans}$	0.72	0.62	0.39	0.48 (±0.01)	0.67	0.54	**0.65**	**0.59*** (±0.02)	0.67	0.54	0.60	0.57* (±0.01)
XLM-R$_{trans}$	**0.74**	0.69	0.42	**0.52** (±0.01)	0.69	0.58	0.55	0.56 (±0.00)	**0.72**	0.62	0.60	**0.61*** (±0.02)

Table 3: The results of our experiment measuring the impact of native hate samples on code-mixed hate detection. The evaluating parameters are accuracy(Acc), precision(Pre), recall(Rec) and F1-score($F1$). The scores under columns **Code-mixed**, **Train-1**, and **Train-2** reported the performance when training was done with the original code-mixed training set, and additionally formulated training sets explained in section 3.2. The variations in $F1$ scores for different random seeds were mentioned in brackets.

code-mixed contexts. Figure 1 reported some of such cases. MLMs gave more attention scores to words like 'murder' and 'rape' when they were trained with additional native samples.



(a) XLM$_{trans}$

(b) mBERT$_{trans}$

Figure 1: Visualisation of attention scores: The upper row in each sub-figure reported the attention scores when training was done only on code-mixed samples, whereas the lower row reported the attention scores when training was done with **Train-1**. We also observed similar patterns for models trained on **Train-2**.

4.2 Exp.-2 observations:

Here, we presented the results of our experiments quantifying the impact of native samples when they were added in different amounts. As mentioned in section 3.3, we created many training sets by incrementally adding native sample batches in the original code-mixed training set. The $F1$ scores obtained for different training sets with different label ratios were presented in Figure 2. The absolute values of all measured parameters were reported in the supplementary material. Some of the interesting observations we got are,

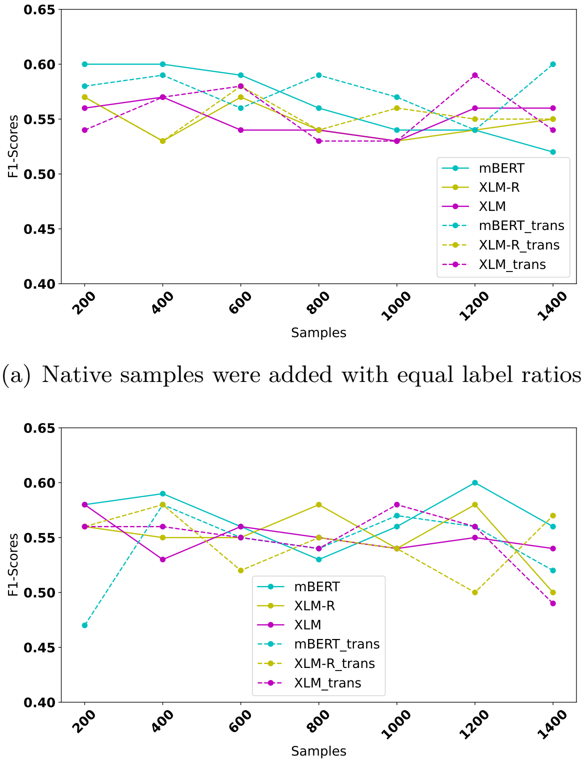
- There was no major change in the $F1$-scores when we added native samples in different amounts. This observation holds irrespective of when samples were added with label ratios similar to **Train-1** or **Train-2**.

- While the $F1$ score stayed between **0.52** and **0.6** for equal label ratio set-up, a relatively high fluctuation, i.e. between **0.47** and **0.6** was observed when label ratio was kept in proportional to the original code-mixed training set.

4.3 Exp.-3 observations:

In this section, we reported the results of our experiment evaluating the performance of MLMs when they were trained with only native samples. As described in section 3.4, we trained the MLMs using (i) Hindi samples, (ii) English samples, and (iii) both English and Hindi samples together. We kept the equal label ratios in all of the training sets. This is because we didn't observe any significant $F1$-score variations in the previous experiments by not keeping the same. The $F1$-scores of MLMs for different training sets were reported in Table 4. We observed the following,

- The highest $F1$-score we got when models were trained using only English samples



(a) Native samples were added with equal label ratios

(b) Native samples were added in a ratio as observed in the original code-mixed training set

Figure 2: Results of Experiments quantifying the impact of native samples when they were added in different amounts. The $x - axis$ represents the native sample batch size, and the $y - axis$ represents the $F1$ scores.

Models	Training sets		
	English	Hindi	English + Hindi
XLM	0.47 (±0.03)	0.34 (±0.04)	0.36 (±0.02)
mBERT	0.50 (±0.04)	0.41 (±0.03)	0.51 (±0.01)
XLM-R	0.53 (±0.01)	**0.60** (±0.03)	**0.60** (±0.01)
XLM$_{trans}$	0.36 (±0.04)	0.21 (±0.05)	0.45 (±0.04)
mBERT$_{trans}$	0.48 (±0.02)	0.31 (±0.04)	0.51 (±0.02)
XLM-R$_{trans}$	**0.54** (±0.00)	0.55 (±0.02)	0.57 (±0.03)

Table 4: $F1$-scores of MLMs when they were trained with only native samples. The variations in $F1$ scores for different random seeds were reported in brackets.

was **0.54**, while the same with Hindi samples raised to **0.60**. We found this interesting, particularly because all Hindi samples were in the Devanagari script, while most code-mixed test samples were in the Roman script. The possible reason behind this could be that both Hindi and code-mixed samples came from the same cultural context. Both datasets were labelled by annotators who know Hindi; hence, the models possibly captured the common cultural context. Some of the past works (2021; 2022) also reported similar phenomena, arguing that cross-lingual learning results by MLMs were good only when training samples and test samples came from similar languages and their cultural context were similar.

- The variants of the XLM-R language model gave the highest $F1$ scores across all native training scenarios; this means the projections of XLM-R generalise better from the native context to the code-mixed context.

- There were no significant improvements in the best $F1$ scores when Hindi and English samples were trained together, compared to when training was done with only Hindi samples. Also, the highest $F1$ score we got by training on **Train-1** or **Train-2** sets was **0.61**, which is a minimal improvement over when training was done with native samples (**0.60**). This empirically points out that MLMs trained on native samples can predict code-mixed hate to a large extent.

5 Error Analysis:

In this section, we reported the cases in which combining the native samples improved or degraded the performance of considered MLMs. Table 5 reported some of such samples. Note that we reported the predictions only for MLMs with additional transformer layers. We see similar performance without transformer layers as well. We observed the following:

- The addition of the native samples helped the MLMs to identify hate, expressed in a code-mixed phrase, without using any explicit hate words. Examples reported in *Sl. No.* 1 and 2 demonstrate this. Even though there is no explicit hate word mentioned in *'tum logo ne hi karwaya tha blast..'* (**Gloss:** You guys have done blast..), most MLMs trained with added native samples could detect the hate. Similarly, in *SL. No. 2*, MLMs trained without native samples failed to understand that the Hindi hate word *'nafrat'* is used to convey a non-hate message.

Sl No	Sample	Translated English	Label	Codemixed			Combined					
				XLM_{trans}	mBERT_{trans}	XLM-R_{trans}	XLM_{trans}		mBERT_{trans}		XLM-R_{trans}	
							T1	T2	T1	T2	T1	T2
1	Tum logon ne hi karwaya tha blasts in PAKISTAN iss liye aise posts daal rahe ho	You guys were the ones who orchestrated the blasts in Pakistan, that's why you're posting such things.	Hate	✗	✗	✗	✓	✗	✓	✓	✓	✗
2	Kabhi nafrat to kabhi dilo ka mail hai,	Sometimes it's hatred, sometimes it's the connection of hearts.	Non-Hate	✗	✗	✗	✓	✓	✓	✓	✓	✓
3	Tabhi Loktantra ka Murder BJP karti hai Supreme court bhi kai bar Phatkaar laga Chuka hai	That's why BJP commits the murder of democracy; the Supreme Court has also reprimanded them many times.	Hate	✗	✗	✗	✗	✗	✗	✗	✓	✗
4	Bhai...Indian bhulakkar hote he ...San bhul hate he...notebandhhi...kya hua?	Brother... Indians are forgetful... They forget everything... Demonetization... what happened?	Hate	✓	✓	✓	✗	✓	✗	✗	✗	✓
5	Abhi bhe time he sab sudar jao.	There's still time. Everyone, mend your ways.	Non-Hate	✓	✗	✓	✗	✓	✗	✓	✗	✓

Table 5: Selected examples for various cases reported under error analysis. Here, the check-mark, and the cross-mark denote correct and incorrect classification by the corresponding model, respectively. Notation: T1 for Train-1 and T2 for Train-2. The columns under **Codemixed** reported the results when the models were trained with only code-mixed samples.

- For the cases where hate is expressed in a sarcastic tone, we observed that MLMs generally struggle to identify it. For example, in *Sl. No.* 3, only the XLM-R model trained **T1** could identify it.

- Finally, in many cases, like *Sl. No.* 5 and 6, we saw performance degradation after adding the native samples. After a careful inspection by the expert linguists, we found that they are some of the hard cases to identify, even for them. This is because the hate expressed here is subjective. For instance, if we see the translation of *Sl. No.* 5 i.e. *"Brother... Indians are forgetful... they forget everything... demonetization... what happened?"*, the label assigned to it was 'Hate'; however, many linguists preferred to categorise it as a 'criticism' without having a solid hate component (E.g. if the writer is an Indian). Similarly, the sample in row 6 translates to *'there is still time, improve yourself'* can be considered a case of implicit hate (patronizing and condescending language) depending on the context.

6 Conclusions and Future Works:

Conclusion: In this paper, we presented several experiments to evaluate the impact of native language samples on code-mixed hate detection. Some of the important observations we got were,

- On combining the native hate samples in the code-mixed training set, we found that MLMs performed relatively better. For many of them, the improvements were statistically significant. The attention scores produced by MLMs also validated this fact.

- We found no major difference when native samples were mixed with an equal label ratio or the ratio as observed in the code-mixed training set.

- We didn't observe any significant F1-score variations when native samples were added incrementally. The F1-score fluctuated between **0.47** and **0.6** for all the cases.

- One interesting observation was that MLMs trained with only native samples could identify hate in the code-mixed context to a large extent. It implies that we can deploy models trained on native samples if an appropriate code-mixed corpus is unavailable. Further, we also observed that the native language datasets that share maximum cultural overlap with the code-mixed hate corpus played a more prominent role in capturing code-mixed hate.

- Finally, in error analysis, we saw that for the cases where code-mixed hate was sarcastic and where hate was subjective, it's still hard to identify them by simply including native hate samples.

Future directions: Our work opened many future directions, such as,

- In this paper, we experimented with only the Hindi-English code-mixed scenario. In future, one can check if similar results replicate for other code-mixed language pairs.
- It could be interesting to see if including native samples from other sentiment tasks, such as humour detection and sarcasm detection in a multi-task framework, could improve code-mixed hate detection.

7 Related works

7.1 Monolingual hate:

In the monolingual set-up, detecting and mitigating hate and similar components were extensively studied in the literature (2023; 2023; 2023). Still, as mentioned earlier, the language coverage is yet to rise significantly. From a task formulation point of view, we saw several variations, such as binary, multi-class, and multi-label classification. The details of datasets under various task-formulations were reported in Appendix B. From an architecture point of view, over the years, we observed a general shift from traditional rule-based and keyword-based systems to machine-learning methods (SVM, Random forest, etc.) to deep-neural methods(linear layers, RNNs, LSTMs, CNNs, Transformers)(2023; 2023; 2023). With the rise of transformer-based language models, fine-tuning their last layers or using neural layers on top of them got approved as a standard go-to approach. Meanwhile, in parallel, researchers also experimented with strategies like ensemble methods, multi-tasking and transfer learning (2023; 2023; 2023) frameworks. These methods produced state-of-the-art results in their respective timelines. For model explainability, people focused on projecting attention scores on input segments to check the model focus on hate words and hate targets (2021).

7.2 Code-mixed Hate:

Very few studies focused on hate detection in code-mixed environments (2018; 2018; 2018b). Most of them focused on the Hindi-English code-mixed scenario. Note that we did not consider studies like (2020) as code-mixed because the samples were only transliterated to another language (Urdhu samples in the Roman script in this case). From a task formulation point of view, we also saw several variations. A detailed report of datasets under various task-formulations reported in Appendix B. Approach-wise, we see people have relied upon (i) feature-based methods (n-grams, lexicons, negations etc.) with traditional machine learning algorithms like SVM, Random forest etc. (2018b), and (ii) neural architectures like CNNs, and LSTMs on top of LIWC features and sentiment scores (2018).

7.3 Hate detection by cross-lingual learning:

In the context of using cross-lingual learning for hate detection, Bigoulaeva et al. (2021) demonstrated that neural architectures like CNNs and LSTMs, without using pre-trained MLMs, could detect hate expressed in German text while training was done with the English hate corpora. However, they additionally used an unlabelled German corpora. Firmino et al. (2023), on the other hand, showed that training MLMs like mBERT and XLM with English and Italian hate corpora can detect the hate in Portuguese samples. However, as mentioned earlier, previous work reported that cross-lingual learning fails when hate (i) expresses language-specific taboos and (ii) is specific to a community or culture.

7.4 Research gap:

We couldn't find any critical study evaluating cross-lingual learning and MLMs on code-mixed hate detection. We argue that limitations of cross-lingual learning(2021) for hate detection, like failing to identify language-specific taboos and culture-specific hate, may not appear in the code-mixed setup. This is because the native corpora are expected to be capturing both. As an initial attempt to see how cross-lingual learning helps in code-mixed hate detection, we considered a simple binary hate classification framework, i.e. 'hate' or not, in this work.

References

Areej Al-Hassan and Hmood Al-Dossari. 2019. Detection of hate speech in social networks: a survey on multilingual corpus. In *6th international con-*

ference on computer science and information technology, volume 10, pages 10–5121.

Azalden Alakrot, Liam Murray, and Nikola S Nikolov. 2018. Dataset construction for the detection of anti-social behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76. IEEE.

Ika Alfina, Rio Mulia, Mohamad Ivan Fanany, and Yudo Ekanata. 2018. Hate speech detection in the indonesian language: A dataset and preliminary study. In *9th International Conference on Advanced Computer Science and Information Systems, ICACSIS 2017*, pages 233–237. Institute of Electrical and Electronics Engineers Inc.

Irina Bigoulaeva, Viktor Hangya, and Alexander Fraser. 2021. Cross-lingual transfer learning for hate speech detection. In *Proceedings of the first workshop on language technology for equality, diversity and inclusion*, pages 15–25.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018a. A dataset of Hindi-English code-mixed social media text for hate speech detection. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 36–41, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018b. A dataset of hindi-english code-mixed social media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Anusha Chhabra and Dinesh Kumar Vishwakarma. 2023. A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, pages 1–28.

Yi-Ling Chung, Elizaveta Kuzmenko, Serra Sinem Tekiroğlu, and Marco Guerini. 2019. Conan-counter narratives through nichesourcing: a multilingual dataset of responses to fight online hate speech. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2829.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Mithun Das, Punyajoy Saha, Binny Mathew, and Animesh Mukherjee. 2022. Hatecheckhin: Evaluating hindi hate speech detection models. *arXiv preprint arXiv:2205.00328*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):1–30.

Anderson Almeida Firmino, Cláudio de Souza Baptista, and Anselmo Cardoso de Paiva. 2023. Improving hate speech detection using cross-lingual learning. *Expert Systems with Applications*, page 121115.

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

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 260–266.

Jennifer Golbeck, Zahra Ashktorab, Rashad O Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A Geller, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, et al. 2017. A large labeled corpus for online harassment research. In *Proceedings of the 2017 ACM on web science conference*, pages 229–233.

Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1470–1478.

Jarod Govers, Philip Feldman, Aaron Dant, and Panos Patros. 2023. Down the rabbit hole: Detecting online extremism, radicalisation, and politicised hate speech. *ACM Computing Surveys*.

Vikram Gupta, Sumegh Roychowdhury, Mithun Das, Somnath Banerjee, Punyajoy Saha, Binny Mathew, Animesh Mukherjee, et al. 2022. Multilingual abusive comment detection at scale for indic languages. *Advances in Neural Information Processing Systems*, 35:26176–26191.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, page 126232.

Akshita Jha and Radhika Mamidi. 2017. When does a compliment become sexist? analysis and classification of ambivalent sexism using twitter data. In *Proceedings of the second workshop on NLP and computational social science*, pages 7–16.

Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of hindi-english code-mixed data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018. Did you offend me? classification of offensive tweets in hinglish language. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, pages 138–148.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the first workshop on abusive language online*, pages 52–56.

Debora Nozza. 2021. Exposing the limits of zero-shot cross-lingual hate speech detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 907–914.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4675–4684.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55:477–523.

Rahul Pradhan, Ankur Chaturvedi, Aprna Tripathi, and Dilip Kumar Sharma. 2020. A review on offensive language detection. *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*, pages 433–439.

Jing Qian, Anna Bethke, Yinyin Liu, Elizabeth Belding, and William Yang Wang. 2019. A benchmark dataset for learning to intervene in online hate speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4755–4764.

Mohammadreza Rezvan, Saeedeh Shekarpour, Lakshika Balasuriya, Krishnaprasad Thirunarayan, Valerie L Shalin, and Amit Sheth. 2018. A quality type-aware annotated corpus and lexicon for harassment research. In *Proceedings of the 10th acm conference on web science*, pages 33–36.

Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in roman urdu. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 2512–2522.

Bjorn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In *3rd Workshop on Natural Language Processing for Computer-Mediated Communication/Social Media*, pages 6–9. Ruhr-Universitat Bochum.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian twitter corpus of hate speech against immigrants. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).*

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.

Nicolas Tsapatsoulis and Vasiliki Anastasopoulou. 2019. Cyberbullies in twitter: A focused review. In *2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 1–6. IEEE.

Pooja Wadhwa and MPS Bhatia. 2013. Tracking on-line radicalization using investigative data mining. In *2013 National Conference on Communications (NCC)*, pages 1–5. IEEE.

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

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.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399.

A Dataset examples

Classes	English-Hindi code-mixed
Hate	Aise logo se sakht nafrat karta hu Jo caste ko naam ke sath jod ke chaude hote h but real me vo piddu hote h **Translation:** I strongly hate those people who associate caste with their names and act tough, but in reality, they are weak.
Non-Hate	Sir phansi nahi. .. sirf looted money wapas chaiya **Translation:** Sir, not execution, we just want the looted money back.

Classes	English
Hate	I don't know how much more I can take! 45 is a compulsive liar! #Trump30Hours #TrumpIsATraitor
Non-Hate	Stand by the doctors , with the doctors. They need support and world needs to interfere.

Classes	Hindi
Hate	दलित बिलकुल नीच होते हैं **Translation:** Dalits are downright mean.
Non-Hate	हमें महिलाओं का प्रोत्साहन करना चाहिए **Translation:** We should encourage women.

Table 6: Samples from all three considered datasets. The Hindi parts (Romanized or Devanagri) are underlined for identification. We have provided English translation for code-mixed and Hindi samples for readability.

B Details of related works:

B.1 Monolingual setup:

There were several datasets proposed that did binary categorization of samples such as: 'hate' or not, (2017; 2019; 2018), 'harassment' or not (2017), 'personal attack' or not (2017), 'sexism' or not (2017), 'offensive' or not (2018), 'anti-refugee hate' or not (2016; 2018) and 'islamophobic' or not (2019). Similarly, from a multi-class and multi-label perspective, researchers combined previously mentioned labels; datasets were annotated with: 'sexist'/'racist'/'none' (2016), 'obscene'/'offensive but not obscene'/'clean' (2017), hate against 'gender'/'sexual orientation'/'religion/'disability'/'none' (2020; 2018; 2019) etc.

B.2 Code-mixed setup:

In binary classification framework, Bohra et al. (2018b) presented a dataset of 4.8k tweets, each annotated with 'hate'/'non-hate'. In multi-class and multi-label set-ups Kumar et al. (2018) published a dataset of 21k Facebook posts, each annotated with one of three labels ('covert aggression'/ 'overt aggression'/ 'none') and their associated category ('physical threat', 'sexual threat', 'identity threat', 'non-threatening aggression'). Similarly, Mathur et al. (2018) produced a dataset focusing only on sexism tweets, each annotated with 'not-offensive'/ 'abusive'/ 'hate'.

C Performance report for Exp 2 and 3:

The absolute values of performance measuring parameters for different set-ups considered under Experiments 2 are shown in Table 7 .

Models	**Ratio**	**Scores**	**Samples size**						
			200	**400**	**600**	**800**	**1000**	**1200**	**1400**
mBERT	Equal ratio	ACC	0.68	0.70	0.71	0.71	0.69	0.66	0.65
		F1	0.60	0.60	0.59	0.56	0.54	0.54	0.56
		PRE	0.56	0.58	0.62	0.62	0.59	0.53	0.51
		REC	0.64	0.63	0.56	0.50	0.49	0.55	0.53
	CM ratio	ACC	0.69	0.67	0.68	0.73	0.72	0.72	0.68
		F1	0.58	0.59	0.56	0.53	0.56	0.60	0.56
		PRE	0.56	0.54	0.56	0.71	0.66	0.61	0.57
		REC	0.60	0.64	0.55	0.42	0.49	0.60	0.56
XLM-R	Equal ratio	ACC	0.71	0.70	0.68	0.70	0.70	0.69	0.68
		F1	0.57	0.53	0.57	0.54	0.53	0.54	0.55
		PRE	0.61	0.60	0.56	0.62	0.62	0.58	0.56
		REC	0.54	0.48	0.58	0.49	0.46	0.50	0.53
	CM ratio	ACC	0.68	0.71	0.70	0.70	0.69	0.66	0.69
		F1	0.56	0.55	0.55	0.58	0.54	0.58	0.50
		PRE	0.56	0.62	0.60	0.59	0.59	0.52	0.60
		REC	0.57	0.50	0.50	0.57	0.49	0.64	0.43
XLM	Equal ratio	ACC	0.66	0.67	0.69	0.70	0.66	0.67	0.67
		F1	0.56	0.57	0.54	0.54	0.53	0.56	0.54
		PRE	0.53	0.54	0.57	0.60	0.53	0.54	0.54
		REC	0.59	0.59	0.52	0.49	0.53	0.58	0.58
	CM ratio	ACC	0.69	0.70	0.68	0.68	0.69	0.70	0.68
		F1	0.58	0.53	0.56	0.55	0.54	0.55	0.54
		PRE	0.57	0.61	0.56	0.56	0.58	0.61	0.56
		REC	0.59	0.46	0.57	0.55	0.50	0.51	0.53
mBERT$_{trans}$	Equal ratio	ACC	0.69	0.70	0.66	0.69	0.66	0.68	0.69
		F1	0.58	0.59	0.56	0.59	0.57	0.54	0.60
		PRE	0.56	0.59	0.52	0.57	0.52	0.57	0.56
		REC	0.61	0.59	0.59	0.61	0.63	0.51	0.64
	CM ratio	ACC	0.71	0.69	0.69	0.71	0.69	0.65	0.70
		F1	0.47	0.58	0.55	0.54	0.57	0.56	0.52
		PRE	0.68	0.56	0.58	0.65	0.58	0.52	0.63
		REC	0.36	0.60	0.51	0.47	0.55	0.61	0.45
XLM-R$_{trans}$	Equal ratio	ACC	0.68	0.70	0.69	0.70	0.70	0.70	0.69
		F1	0.57	0.53	0.58	0.54	0.56	0.55	0.55
		PRE	0.55	0.61	0.56	0.61	0.60	0.60	0.58
		REC	0.59	0.47	0.61	0.48	0.52	0.51	0.52
	CM ratio	ACC	0.69	0.69	0.70	0.70	0.67	0.70	0.69
		F1	0.56	0.58	0.52	0.55	0.54	0.50	0.57
		PRE	0.58	0.57	0.63	0.61	0.54	0.65	0.57
		REC	0.54	0.60	0.44	0.51	0.54	0.40	0.57
XLM$_{trans}$	Equal ratio	ACC	0.68	0.71	0.68	0.69	0.71	0.68	0.67
		F1	0.54	0.57	0.58	0.53	0.53	0.59	0.56
		PRE	0.56	0.61	0.55	0.60	0.63	0.55	0.56
		REC	0.53	0.53	0.62	0.47	0.46	0.63	0.52
	CM ratio	ACC	0.70	0.70	0.69	0.68	0.69	0.70	0.65
		F1	0.56	0.56	0.55	0.54	0.58	0.56	0.49
		PRE	0.60	0.61	0.58	0.56	0.57	0.59	0.52
		REC	0.53	0.51	0.52	0.52	0.59	0.53	0.46

Table 7: Results of experiment-2