Revealing the impact of synthetic native samples and multi-tasking strategies in Hindi-English code-mixed humour and sarcasm detection

Anonymous ACL submission

Abstract

In this paper, we reported our experiments with various strategies to improve code-mixed humour and sarcasm detection. Particularly, we tried three approaches: (i) native sample mixing, (ii) multi-task learning (MTL), and (iii) prompting and instruction finetuning very large multilingual language models (VMLMs). In native sample mixing, we added monolingual task samples to code-mixed training sets. In MTL learning, we relied on native and code-mixed samples of a semantically related task (hate detection in our case). Finally, in our third approach, we evaluated the efficacy of VMLMs via few-shot context prompting and instruction finetuning. Some interesting findings we got are (i) adding native samples improved humor (raising the F1-score up to **6.76%**) and sarcasm (raising the F1-score up to **8.64%**) detection, (ii) training MLMs in an MTL framework boosted performance for both humour (raising the F1-score up to **10.67%**) and sarcasm (increment up to **12.35%** in F1-score) detection, and (iii) prompting and instruction finetuning VMLMs couldn't outperform the other approaches. Finally, our ablation studies and error analysis discovered the cases where our model is yet to improve. We provided our code for reproducibility.

1 Introduction:

Humour and sarcasm are complex and subjective emotions that impact the nature of human communication. They can appear in different forms such as exaggeration, dark humour, gross humour, adult or slang expression, insult, offence, etc. (Frenda et al., 2018; Ahuja, 2019). Past study (Bleakley and Sailofsky, 2023) highlighted how they can affect politics amid tragedy. Detecting Humour and Sarcasm becomes more challenging in a code-mixed setting. This is because models now need to understand humour and sarcasm in an utterance expressed through altering multiple languages. More details on the phenomenon of code-mixing are presented in the Appendix. An example of humorous and sarcastic expression in Hindi-English code-mixed language is given in the following. More examples are presented in Figure 1 of Appendix C. In the following example, the English parts are marked in red, and the Hindi parts are marked in blue. We have provided their translations for readability.

- **Humor:** Never take a moral high ground. Wahan railing nahi hai aur kabhi bhi gir sakte hain.
  (**Translation:** Never take a moral high ground. There are no railings and one can fall at any time.)

- **Sarcasm:** Kuch logo ka number iss liye save krte hain ki galti se uth naa jaye.... #sarcasm
  (**Translation:** Some people save their numbers so that they don't get called by mistake.... #sarcasm)

The NLP community has shown significant interest in monolingual humour and sarcasm detection (Abulaish et al., 2020; Joshi et al., 2017, 2020). Unfortunately, there is relatively less focus on the code-mixed settings (Singh and Sharma, 2023; Elayan et al., 2022; Chen et al., 2024a). Therefore, we have few publicly annotated code-mixed corpora available, further acting as a bottleneck in developing new models (Sitaram et al., 2019; Doğruöz et al., 2021; Winata et al., 2023). The evolution of multilingual large language models (MLM hereafter) has shown a new path to address this issue. They can learn task-specific knowledge from samples in one language and make predictions

for samples in different languages. This phenomenon is known as cross-lingual learning. It is very effective if training and testing samples share similar linguistic and cultural contexts (Bigoulaeva et al., 2021; Gupta et al., 2022). MLMs learn their embeddings from a corpora spanning multiple languages, thus they are aware of vocabulary of multiple languages. In the context of code-mixed languages, it means we can fine-tune the MLMs using native monolingual task samples (e.g. for Hindi and English task samples for a Hindi-English code-mixed task) and do prediction for code-mixed samples. The hypothesis is that since code-mixed corpora (in Hindi-English code-mixed language) and the native monolingual corpora (in Hindi and English languages) are likely to share similar linguistic and cultural contexts, the training with native task samples or adding them in code-mixed training sets can improve code-mixed task performance. In fact, Mazumder et al. (2024), in a set of empirical experiments, have shown that adding Hindi and English hate samples in code-mixed hate training corpora improves code-mixed hate detection (Mazumder et al., 2024). However, nobody has tested it for code-mixed Humour and Sarcasm detection. A detailed discussion of prior works in this direction is presented in the Appendix A. Further, we observed that Hindi humour and sarcasm datasets are not publicly available. We experimented with synthetic Hindi samples and multi-tasking strategies to fill this gap. Overall, we asked for three research questions,

R1: *Does mixing native samples (English and synthetic Hindi samples in our case) in code-mixed training sets improve code-mixed humour and sarcasm detection?*

R2: *Do jointly training with a semantically related third task (hate detection) along with native sample mixing improve code-mixed humour and sarcasm detection?*

R3: *Do adding native samples in the prompting context or in instruction finetuning of Very large MLMs (VMLMs hereafter) improve the performance?*

In summary, our contributions are the following,

- We analyzed the effect of adding native samples (both English and synthetic Hindi samples) from existing humour and sarcasm datasets to code-mixed training data. For this, we experimented with two types of models: (i) statistical classifiers on top of word n-gram features, (ii) MLMs such as mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020), MuRIL (Khanuja et al., 2021) and IndicBERT (Doddapaneni et al., 2023) (refer **Exp. 1**: Section 3.2). Combining native samples with code-mixed data led to improvements in MLMs, achieving increment up to **6.76%** and **8.64%** for humor and sarcasm detection, respectively ($p < 0.05$). In contrast, statistical models performed worse.

- We integrated a related task which is hate detection with native samples in a multitask learning framework (refer **Exp. 2**: Section 3.3). Instead of sequential training, the model processed batches containing mixed-task samples. We conducted an ablation study to understand the role of the gating mechanism in the MTL framework (in Appendix G.1). This approach gave significant performance improvements for MLMs, with F1-score up to **10.67%** increment in humour and **12.35%** increment in sarcasm ($p < 0.05$). With a gating mechanism, they better handled *shorter contexts* and *misspelt samples*.

- We compared the performance of VMLMs using native and code-mixed examples as few-shots (refer **Exp. 3**: Section 3.4). In the second set-up, we compared the VMLMs' performance when native samples are addded to the code-mixed training set. However, neither type of set-ups could improve the VMLM predictions.

2 Datasets:

In this section, we reported the details of code-mixed and native datasets considered in our study. Please note that we ignored the datasets containing dialogues and multi-modal samples to simplify our task formulation. The list of the datasets and their basic statistical details were reported in Table 1. The dataset examples were illustrated in Figure 1 in the appendix. A brief description of the individual datasets were reported in Appendix C. Apart from the class distribution, we also provided the Kullback-

	Language	Dataset	# $+ve$	# $-ve$	H $+ve$	H $-ve$	KL	IAA
Humor	Code-mixed	Khandelwal et al. (2018)	1759	1192	31.32%	22.98%	1.603	H: 0.821 & NH: 0.794 (Fleiss' Kappa)
	English	Col(Annamoradnejad and Zoghi, 2024)	100000	100000	64.92%	39.85%	2.386	N/A
	English	POTD(Yang et al., 2015)	2423	2403	47.13%	53.05%	1.318	N/A
	English	HaHa(Meaney et al., 2021)	6179	3821	73.54%	60.64%	1.489	0.736 (Krippendorff's)
	English	16000(Mihalcea and Strapparava, 2005)	16000	16000	49.94%	52.55%	1.209	N/A
Sarcasm	Code-mixed	Swami et al. (2018)	504	4746	31.15%	38.33%	3.021	0.79 (Cohen's Kappa)
	English	NHD(Misra and Arora, 2019)	11724	14985	44.20%	39.26%	1.742	N/A
	English	iSarc(Oprea and Magdy, 2020)	1067	4668	60.44%	51.82%	1.184	N/A
	English	SC-V2(Oraby et al., 2016)	4693	4693	77.69%	79.90%	0.645	0.80
Hate	Code-mixed	Bohra et al. (2018)	1661	2914	71.82%	77.62%	1.218	0.982 (Cohen's Kappa)
	Hindi	HCHIn(Das et al., 2022)	3338	1416	-	-	0.740	0.95 (Fleiss' Kappa)
	English	HASOC[8]	2261	3591	76.47%	60.68%	1.182	72% (overlap)

Table 1: Dataset statistics. Notation: # for number of samples, H denotes proportion of samples containing hurtful (offensive, aggressive and hateful) keywords from positive or negative class, IAA for Inter-Annotator Agreement and KL for symmetrized smoothed Kullback-Leibler divergence between word distributions of positive ($+ve$) and negative ($-ve$) samples.

Leibler (KL) divergence (Kullback and Leibler, 1951) values for the individual datasets in Table 1. The KL divergence measures the difference between two probability distributions. Here, it quantifies lexical variation, i.e., *how the word distribution in class differs from the other*. A higher value of KL divergence suggests that the classes are well-separated in terms of word distributions. We also reported the fraction of total samples containing English hurtful (offensive, aggressive and hateful) keywords in each class. We used the lexicon given by Bassignana et al. (2018) to calculate it. Finally, we also reported the inter-annotator agreement (IAA) scores published with individual datasets.

3 Experiments:

In this section, we reported the details of our experiments conducted as a part of this study. Apart from reproducing the baselines (section 3.1), we designed three experiments each based on a unique research philosophy. The three philosophies are i) native sample mixing (section 3.2), ii) multi-task learning (section 3.3) and iii) prompting and instruction finetuning very large multilingual language models (VMLMs) (section 3.4). While the native sample mixing strategy intends to improve the code-mixed tasks by adding monolingual native samples to the code-mixed training sets, the multi-task learning strategy tries to do the same by learning linguistic knowledge from the samples of a third task (here, it is hate detection). Our last strategy, i.e. prompting and instruction finetuning VMLMs, evaluates the performance of very large multilingual language models for the considered code-mixed tasks in a few-shot context prompting and instruction finetuning scenarios. The detailed experimental set up is given in Appendix F.1.

3.1 Baselines:

In this section, we reported the previously proposed best performing methods as baselines. They were proposed for code-mixed humour and sarcasm detection in Hindi-English code-mixed scenario. Please note that some baseline papers did not share their code; thus, we reimplemented them to the best of our knowledge. Further, we made our codebase public for reproducibility. In the results section, we reported both the reproduced and original results as per the baseline paper. However, we considered the reproduced results for comparative analysis.

Humor: We considered two previous works published by Agarwal and Narula (2021) and Muttaraju et al. (2022) as our baselines. Further details of individual approaches are provided in Appendix - D.1.

Sarcasm: We considered two previous works published by Pandey and Singh (2023) and Aloria et al. (2023) as our baselines. Further details of individual approaches are provided in Appendix - D.2.

3.2 Exp. 1- Impact of mixing native language samples:

Our first experiment explored the impact of native language samples after adding them to the code-mixed training sets. Past study (Mazumder et al., 2024) reported that this strategy works for code-mixed hate detection. However, no one tested it for code-mixed humour

and sarcasm detection. Further, in our case, even though there are publicly available English humour and sarcasm datasets (Table 1), we couldn't find any Hindi datasets for the same. Thus, we choose to create silver annotated datasets by translating some portions of English datasets into Hindi using Google Translator API[1]. Note that accurately translating humour and sarcasm samples is still an open research topic. Thus, we used the most popular publicly available translation tool of current time. From the methodological point of view, we considered two types of models.

- **Statistical classifiers:** We considered three statistical classifiers, i.e. Naive Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM) in our study. NB is known to perform better when there is high KL divergence between classes. We utilized word-level unigrams, bigrams, and trigrams as features. Past works showed these features to perform best (Khandelwal et al., 2018; Swami et al., 2018).

- **Multilingual Language Models (MLMs):** We also considered four widely used MLMs for our study. They are, i) mBERT (Devlin et al., 2019), ii) XLM-R (Conneau et al., 2020), iii) MuRIL (Khanuja et al., 2021) and iv) IndicBERT (Doddapaneni et al., 2023). Out of them, mBERT and XLM-R are general-purpose MLMs trained on 100+ languages, while MuRIL and IndicBERT are specialized models specifically trained for Indic languages. We froze all but the last four layers of MLMs during fine-tuning.

3.3 Exp. 2- Multi-task learning:

In our second approach, we explored the efficiency of multi-task framework to detect code-mixed humour and sarcasm by learning linguistic knowledge from the samples of a third task, i.e., here it is, hate detection. We chose hate detection as a third task because (i) it is semantically related to humour and sarcasm, (ii) samples of humour and sarcasm datasets contained hateful keywords (refer Table 1), and (iii) we could found Hindi, English and code-mixed samples available for this task. Our multi-task framework is inspired by the framework proposed by Rotman and Reichart (2022). They utilized a BERT-like architecture divided into two parts. The bottom module consists of eight lower layers of a transformer-based language model like BERT were common to the participating tasks. The top module, containing top four layers, were separately present for the individual tasks. Apart from that there is an additional top module present for parameter sharing. A gating mechanism connects all of the task-specific top modules with the additional top module. Authors experimented this framework on tasks like dependency parsing and named entity recognition and found that it performs better. In our case (i) we added a regularization term for soft parameter sharing of the final layers from the top module, and (ii) we froze the parameters of the bottom module during fine-tuning. The architecture of our framework is shown in Figure 3. We used embeddings from three widely used MLMs, i.e., (i) mBERT, (ii) XLM-R and (iii) MuRIL to initialize the layers of our MTL framework. We conducted an ablation study (refer Appendix G.1) to examine the role of the gating mechanism within the MTL models by removing the gate and comparing the performance with models that included the gating mechanism.

3.4 Exp. 3- Impact of in-context learning on very large multilingual language models:

In our third experiment, we evaluated the performance of very large multilingual language models in detecting humour and sarcasm in code-mixed texts. As their name suggests, they have a lot of parameters, thus, they are designed to work well with prompting approaches rather than fine-tuning them fully. We utilized four VMLMs: (i) Gemma (Team et al., 2024), (ii) Aya Expanse (Üstün et al., 2024), (iii) Llama 3.1 (Dubey et al., 2024) and (iv) GPT-4[2] (Achiam et al., 2023). These VMLMs were chosen for their strong performance in both Indic languages (Watts et al., 2024) and English. We used two scenarios: (i) few-shot prompting and (ii) instruction finetuning using LoRA (Hu et al., 2021) adapter. Here in the few shot set-up, we show a few training examples

[1] https://cloud.google.com/translate?hl=en

[2] https://chatgpt.com/

in the prompt while asking the VMLMs to classify for a test sample. This strategy has proven its superiority in sentiment-related code-mixed tasks(Yadav et al., 2024). The prompt template and some of the examples are reported in Appendix E.1. We conducted many variants of this experiment by considering samples from native and code-mixed training set as few shots examples in the prompt. In the second scenario, we finetuned the first three open source VMLMs on the code-mixed training set and then after combining the native language samples with the code-mixed training data.

4 Results and discussion:

As the datasets are not balanced, we evaluated our models and the baselines using the F1-score(*F1*). The reported values are the average of three random seeds over separate runs. In the following subsections, we reported the results of our baseline and three experiments. Statistically significant results ($p < 0.05$) are identified with an '*' mark.

4.1 Baseline results:

Most of the baseline papers reported accuracy values as their performance measures. However, since the datasets are class-imbalanced, we believe F1-scores best measures their performance. We implemented the baselines and reported the F1-scores in Table 2. Out of all methods, we found that IndicBERT performed best for humour detection. Similarly, LSTM-BERT gave the best result for sarcasm detection.

	Baselines	F1
Humor	Agarwal and Narula (2021)	**0.78**†
	Muttaraju et al. (2022)	0.74†
Sarcasm	Pandey and Singh (2023)	**0.85**† (0.92$^{\#}$)
	Aloria et al. (2023)	0.84† (0.84$^{\#}$)

Table 2: Baselines results. Notation: originally reported scores are mentioned with '#' mark and our reproduced results are marked as '†'.

4.2 Observations from Exp. 1:

In this section, we reported our observations from the first experiment. The F1-scores obtained from all models over the considered datasets and training scenarios are reported in Table 3. The F1-scores of best-performing models for the individual training scenarios (column-wise in Table 3) were marked in bold, while the second-best results are underlined. Similarly, the best-performing scenarios giving the highest $F1$ scores for individual models (row-wise in Table 3) were kept inside the parenthesis, while the second best scores are marked with '#' superscript. The highest score for both tasks across all training scenarios and models are marked in blue. Following are our takeaways,

- When trained with only code-mixed samples, mBERT gave the highest F1-score of **0.78** for humour detection, followed by XLM-R and MuRIL (F1-score of **0.75**). Similarly, for sarcasm detection, MuRIL reported the highest F1-score of **0.83** followed by IndicBERT and XLM-R (F1-score of **0.81**).

- On adding native samples to the code-mixed training sets, statistical classifiers did not show any performance improvement. In fact, in many cases, the performance declined sharply (a decline of **9.85%** and **56.7%** in F1-scores for humour and sarcasm detection, respectively).

- Among the MLMs, mBERT showed significant improvement after native sample mixing with an F1-score of **0.84** (improvements up to **5%**) in detecting sarcasm. In the case of MuRIL, we saw many instances where it resulted in statistically significant ($p < 0.05$) improvement (up to **5.3%** and **7.2%** raise in F1-score for humour and sarcasm detection respectively) after native sample mixing. Finally, we observed that IndicBERT model performed best after native sample mixing with an F1 scores of **0.79** (improvement up to **6.7%**) and **0.88** (improvement up to **8.6%**) in code-mixed humour and sarcasm detection respectively (both statistically significant with $p < 0.05$).

- We observed that MuRIL and IndicBERT gave the overall highest scores. This is interesting as both are special LLMs exclusively developed for Indian languages. This observation is consistent with Mazumder et al. (2024) as the same phenomenon was observed for code-mixed hate detection. Another important observation

Humor

NLD →		Col			POTD			HaHa			16000		
Model ↓	CM	CM+Hi+En	CM+En	CM+Hi	CM+Hi+En	CM+En	CM+Hi	CM+Hi+En	CM+En	CM+Hi	CM+Hi+En	CM+En	CM+Hi
NB	0.74$^{\#}$	(0.75)	**(0.75)**	(0.75)	(0.75)	(0.75)	(0.75)	0.74$^{\#}$	0.74$^{\#}$	(0.75)	(0.75)	(0.75)	(0.75)
RF	0.72$^{\#}$	0.69	0.70	0.69	0.70	0.72$^{\#}$	0.71	0.71	0.70	(0.73)	0.69	0.69	0.68
SVM	(0.71)	0.68	0.64	0.66	0.69	0.70$^{\#}$	0.69	0.69	0.69	0.69	0.68	0.69	0.69
mBERT	**(0.78)**	0.73	0.72	0.65	0.71	0.76	0.75	0.68	0.76$^{\#}$	0.76$^{\#}$	0.74	0.75	0.70
XLM-R	0.75$^{\#}$	0.72	0.73	0.73	0.73	0.75$^{\#}$	0.73	0.74	0.75$^{\#}$	0.73	**(0.76)**	(0.76)	0.72
MuRIL	0.75	(**0.79***)	0.72	**0.76**	0.73	(**0.79***)	**0.78**$^{*\#}$	0.77	**0.77**	**0.78**$^{\#}$	**0.76**	**0.77**	**0.78**$^{\#}$
IndicBERT	0.74	0.76	0.72	0.71	**0.77**$^{\#}$	0.76	0.75	(**0.79***)	0.76	0.75	0.74	0.73	0.76

Sarcasm

NLD →		NHD			iSarc			SC-V2		
Model ↓	CM	CM+Hi+En	CM+En	CM+Hi	CM+Hi+En	CM+En	CM+Hi	CM+Hi+En	CM+En	CM+Hi
NB	(0.74)	0.35	0.37	0.41	0.38	0.41	0.42$^{\#}$	0.32	0.34	0.37
RF	(0.69)	0.43	0.51	0.60$^{\#}$	0.51	0.57	0.57	0.49	0.60$^{\#}$	0.58
SVM	(0.74)	0.59	0.59	0.73$^{\#}$	0.64	0.64	0.71	0.68	0.68	0.73$^{\#}$
mBERT	0.80	0.83$^{*\#}$	0.79	0.82*	0.79	0.81	0.78	0.78	0.82	(0.84*)
XLM-R	0.81$^{\#}$	(0.83*)	0.79	(0.83*)	0.81$^{\#}$	0.81$^{\#}$	0.81$^{\#}$	0.80	(0.83)	0.81$^{\#}$
MuRIL	**0.83**	**0.86***	(**0.89***)	0.82	**0.84**	0.85*	**0.84**	**0.87**$^{*\#}$	**0.84**	**0.87**$^{*\#}$
IndicBERT	0.81	**0.86**$^{*\#}$	0.86$^{*\#}$	**0.85***	0.81	(**0.88***)	0.83	0.83	0.83	0.82

Table 3: Results of our experiment evaluating the impact of mixing native samples for humor (upper table) and sarcasm (lower table) detection. Notation: NLD for native language dataset, CM for code-mixed. Reported scores are $F1$ scores of the positive class and are averaged over three different random seeds.

was that the addition of Hindi samples didn't result in a sharp improvement in the F1-score as we saw in Mazumder et al. (2024). On inspection, we found that the synthetic samples are not humorous and sarcastic compared to their original English ones. In other words, the humour and sarcasm got lost during translation. A detailed discussion of the same is reported in Appendix H.

4.3 Observations from Exp. 2:

In this section, we presented our observations from the second experiment. The F1-scores obtained from all models for the considered datasets and training scenarios are reported in Table 4. In Table 4, each row enlists different training scenarios for considered two tasks. It has two sub-tables. The upper and lower tables report results for code-mixed humour and sarcasm detection, respectively. The first two columns in each sub-table report the combination of datasets used for training. For example, the first row under the 'Humor' sub-table presents the case where the training set has (i) humour samples from the code-mixed dataset and English dataset 'ColBERT' (written as NLD: 'Col'), (ii) sarcasm samples from the code-mixed dataset and English dataset 'NHD' (tick marked next to NHD). We didn't consider hate samples here (presented as the empty box under the 'Hate' column). So, we trained our models for two tasks in this case. Similarly, the fifth row represents the case where the training set has code-mixed, native English and native Hindi hate samples along with the Humour and Sarcasm samples considered in the first row. So, in this case, we trained our models for three tasks. The overall best scores were marked in red. Following were our takeaways,

- For code-mixed humor detection, MuRIL$_{MTL}$ reported the highest F1 score of **0.83** (up to **10.67%** increment), followed by XLM-R$_{MTL}$ (**0.81**) and mBERT$_{MTL}$ (**0.80**). On the other hand, for code-mixed sarcasm detection, XLM-R$_{MTL}$ outperformed others with **0.91** F1 score (up to **12.35%** increment), followed by MuRIL$_{MTL}$ (**0.88**) and mBERT$_{MTL}$ (**0.86**).

- We achieved the highest scores in MTL strategy when native datasets with *low KL divergence* between classes were combined. This appears to help the pre-trained MLM-based MTL architecture focus more on contextual understanding rather than being influenced by lexical differences between the labels. Notably, SC-V2 and HCHIn, which have the lowest KL divergence, were consistently present among the best-performing configurations in both sub-tables of Table 4 (refer [row 26, col 7] of Humor subtable and [row 32, col 5] of Sarcasm subtable).

- For sarcasm detection, mBERT$_{MTL}$ re-

Humor

NLD : *Col*		mBERT$_{MTL}$		XLM-R$_{MTL}$		MuRIL$_{MTL}$	
Hate	Sarcasm	Gate	w/o Gate	Gate	w/o Gate	Gate	w/o Gate
☐	☑$_{NHD}$	0.69	0.76	0.78*	**0.79**$^{*\#}$	0.76	0.68
☐	☑$_{iSarc}$	0.71	**0.78**	0.76	0.75	0.76	0.71
☐	☑$_{SC-V2}$	0.67	0.76	**0.80**$^{*\#}$	(**0.80***)	0.74	0.76
☑	☐	**0.77**	0.72	0.71	0.75	**0.77**	0.75
☑	☑$_{NHD}$	0.78	**0.79**$^{\#}$	**0.79***	**0.79**$^{*\#}$	0.76	0.78$^{\#}$
☑	☑$_{iSarc}$	0.78	0.77	**0.79***	0.78	**0.79***	0.76*
☑	☑$_{SC-V2}$	0.78	**0.79**$^{\#}$	**0.79***	0.78*	0.75	0.75
NLD : *POTD*							
☐	☑$_{NHD}$	0.71	0.79$^{\#}$	0.79*	(**0.80***)	0.73	0.71
☐	☑$_{iSarc}$	0.76	**0.79**$^{\#}$	0.78	0.78*	0.76	0.78$^{*\#}$
☐	☑$_{SC-V2}$	0.71	0.79$^{\#}$	0.70	(**0.80***)	0.75	0.72
☑	☐	0.69	**0.79**$^{\#}$	0.75	**0.79**$^{*\#}$	0.67	0.70
☑	☑$_{NHD}$	0.79$^{\#}$	0.79$^{\#}$	**0.80**$^{*\#}$	0.79$^{*\#}$	0.74	0.75
☑	☑$_{iSarc}$	**0.79**$^{\#}$	0.78	**0.79***	0.78*	0.78	0.78$^{*\#}$
☑	☑$_{SC-V2}$	0.79$^{\#}$	0.79$^{\#}$	(**0.81***)	0.71*	0.69	0.75
NLD : *HaHa*							
☐	☑$_{NHD}$	0.77	0.78	**0.80**$^{*\#}$	0.78*	0.71	0.76
☐	☑$_{iSarc}$	0.76	0.78	**0.80**$^{*\#}$	0.75	0.69	0.75
☐	☑$_{SC-V2}$	**0.79**$^{\#}$	0.78	0.78	0.78*	0.71	0.75
☑	☐	0.72	0.71	0.77	0.76	**0.80**$^{*\#}$	0.76
☑	☑$_{NHD}$	0.77	0.79$^{\#}$	**0.80**$^{*\#}$	0.79$^{*\#}$	0.79*	0.75
☑	☑$_{iSarc}$	0.78	(**0.80***)	0.77	0.76	0.78	0.78$^{\#}$
☑	☑$_{SC-V2}$	(0.80)	0.79$^{\#}$	0.78*	0.78	(**0.83***)	(0.81*)
NLD : *16000*							
☐	☑$_{NHD}$	0.76	0.68	0.79*	(**0.80***)	0.69	0.76
☐	☑$_{iSarc}$	0.76	0.78	**0.80**$^{*\#}$	0.77	0.78*	0.69
☐	☑$_{SC-V2}$	**0.79**$^{\#}$	0.68	0.78	**0.79**$^{*\#}$	0.73	0.68
☑	☐	**0.78**	**0.78**	0.77	**0.78***	0.77	0.70
☑	☑$_{NHD}$	0.76	**0.79**$^{\#}$	0.77	**0.79**$^{*\#}$	0.78	0.76
☑	☑$_{iSarc}$	**0.78**	0.77	0.77	0.75	**0.78**	0.76
☑	☑$_{SC-V2}$	0.78	(**0.80***)	**0.80**$^{*\#}$	0.77	**0.80**$^{*\#}$	0.77

Sarcasm

NLD : *NHD*		mBERT$_{MTL}$		XLM-R$_{MTL}$		MuRIL$_{MTL}$	
Hate	Humor	Gate	w/o Gate	Gate	w/o Gate	Gate	w/o Gate
☐	☑$_{Col}$	0.82	0.83	**0.85***	0.82	0.83	0.78
☐	☑$_{POTD}$	0.84	0.83	**0.85***	0.82	0.82	0.76
☐	☑$_{HaHa}$	0.84	0.83	**0.85***	0.82	0.84	0.78
☐	☑$_{16000}$	0.81	0.83	**0.90**$^{*\#}$	0.82	0.86*	0.79
☑	☐	0.81	0.85$^{*\#}$	**0.86***	0.82	**0.86**	0.78
☑	☑$_{Col}$	0.83*	(0.86*)	0.84*	**0.88**$^{*\#}$	0.86*	(0.84)
☑	☑$_{POTD}$	0.84*	0.81	**0.88***	0.87*	0.84	0.82
☑	☑$_{HaHa}$	0.84*	0.83*	**0.88***	0.87*	0.81	0.81
☑	☑$_{16000}$	0.85$^{*\#}$	0.85$^{*\#}$	**0.88***	0.82	0.87$^{*\#}$	0.81
NLD : *iSarc*							
☐	☑$_{Col}$	0.81	0.83	0.83*	0.81	**0.86**	0.78
☐	☑$_{POTD}$	0.83*	**0.85**$^{*\#}$	0.83	0.80	0.82	0.79
☐	☑$_{HaHa}$	0.84*	0.81	0.85*	0.81	**0.86**	0.79
☐	☑$_{16000}$	0.81	**0.85**$^{*\#}$	0.82	0.81	0.82	0.78
☑	☐	0.82	0.83*	0.82	0.81	**0.85**	0.72
☑	☑$_{Col}$	0.81	0.80	0.86*	0.82	(**0.88***)	0.79
☑	☑$_{POTD}$	0.82	0.74	**0.88***	0.83	0.82	0.79
☑	☑$_{HaHa}$	0.81	0.81	**0.88***	0.82	0.85	0.80
☑	☑$_{16000}$	0.81	0.81	**0.88***	0.85*	0.82	0.79
NLD : *SC-V2*							
☐	☑$_{Col}$	**0.84***	**0.84***	0.83	0.83	0.81	0.83$^{\#}$
☐	☑$_{POTD}$	0.82	0.82	0.84*	0.83	**0.85**	0.83$^{\#}$
☐	☑$_{HaHa}$	**0.85**$^{*\#}$	0.83*	0.82	0.83	**0.85**	0.83$^{\#}$
☐	☑$_{16000}$	0.81	0.84*	**0.85***	0.83	0.84	0.83$^{\#}$
☑	☐	0.83*	0.84*	**0.88***	0.83	0.85	0.79
☑	☑$_{Col}$	0.80	0.82	**0.89***	0.87*	0.84	0.81
☑	☑$_{POTD}$	0.83*	0.83*	**0.89***	0.85*	0.86*	(0.84)
☑	☑$_{HaHa}$	(0.86*)	0.84*	0.85*	(**0.89***)	0.87$^{*\#}$	0.79
☑	☑$_{16000}$	0.83*	0.81	(**0.91***)	0.86*	0.85	0.81

Table 4: Results of our experiment of multi-task learning and ablation study for humor (upper table) and sarcasm (lower table) detection. Notation: 'NLD' for Native Language Dataset. Reported scores are $F1$ scores of the positive class and are averaged over three different random seeds.

sulted with a highest F1 score of **0.86** (improvement up to **7.5%**). The improvement is statistically significant ($p < 0.05$).

Humor

Model	CM	Col	POTD	HaHa	16000
Gemma	0.09	0.09	0.07	(0.11)	0.10$^{\#}$
Aya Expanse	0.73	(**0.74**)	0.73$^{\#}$	0.71	0.73
Llama-3.1	(**0.75**)	**0.74**$^{\#}$	(**0.75**)	(**0.75**)	(**0.75**)
GPT-4	(0.74)	0.57	0.62$^{\#}$	0.56	0.58

Sarcasm

Model	CM	NHD	iSarc	SC-V2
Gemma	0.34	0.11	0.39$^{\#}$	(**0.47**)
Aya Expanse	0.21	0.21	(0.26)	0.23$^{\#}$
Llama-3.1	0.21	**0.45**$^{\#}$	(**0.51**)	0.28
GPT-4	(**0.78**)	0.17	0.25	0.43$^{\#}$

Table 5: Results of our experiment evaluating the impact of in-context learning on VMLMs. Notation: CM for code-mixed.

- Similarly, for humor detection, XLM-R$_{MTL}$ gave an improvement of F1 score upto **0.81** (improvement up to **8%**). For sarcasm detection, XLM-R$_{MTL}$ improved even more with the highest F1 score of **0.91** (improvement up to **12.35%**). Both the improvements were statistically significant with $p < 0.05$.

- Finally, MuRIL$_{MTL}$ reported the highest F1 score for humor detection, i.e., **0.83** (an improvement up to **10.67%**). The improvement is statistically significant ($p < 0.05$). Similarly, for sarcasm detection, F1 scores ranged from **0.81** to **0.88** using MuRIL$_{MTL}$. The F1-score improved up to **6%** which is statistically significant ($p < 0.05$).

- Upon analyzing the improvements deeply, we found that the MTL models performed better for samples with *shorter context* length (refer to Figure 5 in Appendix G) and *spelling errors*. Due to space constraints, we reported these insights from the ablation study in Appendix G.1.

- The error analysis (refer Appendix G.2) revealed that samples with some connection to the hate detection task improved performance in multitask setting.

4.4 Observations from Exp. 3:

In this section, we reported our observations from the third experiment. The best F1-scores obtained from prompting VMLMs are reported in Table 5. A more detailed overview of obtained F1-scores is presented in Table 7. The F1-scores obtained from instruction finetuning

Humor					
Model	CM	CM+Col	CM+POTD	CM+HaHa	CM+16000
Gemma	**(0.77)**	0.58	**(0.77)**	0.23	**0.76**
Aya Expanse	0.74	**0.74**	0.75	**0.74**	**(0.76)**
Llama-3.1	**(0.77)**	**0.74**	0.75	**0.74**	0.75

Sarcasm				
Model	CM	CM+NHD	CM+iSarc	CM+SC-V2
Gemma	(0.74)	0.63	0.67	0.64
Aya Expanse	0.78	**0.68**	**(0.79)**	0.78
Llama-3.1	**0.80**	0.49	0.70	**(0.81)**

Table 6: Results of our experiment evaluating the impact of native language mixing in instruction finetuning of VMLMs using LoRA adapter. Notation: CM for code-mixed.

VMLMs are presented in Table 6. The highest overall scores for both tasks are marked in blue. Following were our takeaways,

- When we prompted the VMLMs with code-mixed few shots, Llama-3.1 achieved the highest F1-score (**0.75**) for humour detection, followed by GPT-4 (**0.74**) and Aya Expanse (**0.73**). In contrast, Gemma performed the worst with an F1-score of **0.09**. For sarcasm detection, GPT-4 outperformed all models with an F1-score of **0.78**, while others showed a sharp decline, i.e., Gemma (**0.34**), Aya Expanse (**0.21**), and Llama-3.1 (**0.21**).

- When prompted with native humour few-shot examples, Llama-3.1 maintained a stable F1-score between **0.74**–**0.75**, while Aya Expanse showed no significant improvement, maintaining scores in the **0.71**–**0.74** range. In contrast, Gemma performed poorly, with F1-scores ranging from **0.07**–**0.11**. GPT-4 also experienced a decline in performance, with F1-scores dropping to **0.56**–**0.62**. Native few-shot prompting led to some improvements in sarcasm detection across models. Gemma's performance significantly increased to **0.47**, compared to **0.34** with code-mixed few-shots. Similarly, Aya Expanse improved to **0.26**, up from **0.21**, while Llama-3.1 achieved an F1-score of **0.51**, a substantial increase from **0.21** in the code-mixed setting. However, GPT-4 continued to struggle, with F1-scores ranging from **0.17**–**0.43**.

- When VMLMs were finetuned using the code-mixed training set, Gemma and Llama-3.1 achieved the best F1-score of **0.77** in humour detection, while Llama-3.1 got best F1-score of **0.80** in sarcasm detection. When finetuned using native sample mixing strategy, Aya Expanse and Llama-3.1 maintained comparable performances within the range **0.74-0.76**, whereas Gemma showed greater fluctuations (**0.23-0.77**). For sarcasm detection also, VMLMs showed varying fluctuations, i.e., Gemma (**0.63-0.67**), Aya Expanse (**0.68-0.79**), and Llama-3.1 (**0.49-0.81**). Although this finetuning approach outperformed the few-shot prompting method, it did not exceed the best F1-scores obtained with MLMs. Futhermore, a consistent pattern emerged from the native dataset point of view, where *low KL divergence* native datasets proved to be more effective for training compared to others.

5 Conclusion:

From our findings, we drew the following conclusions:

- Among these three strategies, MTL reported the most significant improvement, with F1-score increments upto **10.67%** for humor and **12.35%** for sarcasm. Native sample mixing followed, with increments upto **6.76%** for humor and **8.64%** for sarcasm, while VMLMs in both set-up showed no improvement in F1-scores.

- The ablation study highlighted the importance of the gating mechanism within the MTL framework, particularly in samples with *'shorter context lengths'* and those containing *'misspelled words'*.

- The error analysis (refer Appendix G.2) presented samples which justified the utility of related tasks in the multitask scenario.

- The VMLMs showed poor performance due to a tendency to favor specific labels in sentiment-related classification tasks (refer Appendix E.3 for detailed observations). This insight aligned well with previous works (Góes et al., 2023; Baranov et al., 2023; Zhang et al., 2024) which showed LLMs inability in detecting monolingual humor and sarcasm.

6 Limitation:

In this section, we reported some of the limitations of our work.

- Our MTL models occasionally failed on non-humorous and non-sarcastic samples. The possible reason behind it could be task and domain interference. We reported a detailed qualitative analysis of correctly and incorrectly classified samples in error analysis section (refer Appendix G.2).
- Non-availability of native Hindi language datasets for these two tasks restricted us from utilizing gold labeled Hindi samples for mixing in our experiments.
- Due to the limited linguistic expertise, we evaluated our hypothesis only on the Hindi-English code-mixed scenarios. Other language pairs can be utilized to shed some light on the generalization of our approach to more languages.
- We couldn't make use of more larger VMLMs due to computational constraints. The larger ones (with more than 100B parameters) may generalize the results more clearly.
- Here, in our experiments we utilized the most widely used translator (Google Translate API) to generate synthetic Hindi samples. One can try other possible ways for synthetic data generation using various VMLMs, in the same lines as future scope.
- We explored the impact of native samples for code-mixed humor and sarcasm detection. As future scope, one can test the impact of near native language (languages which have similar origin) samples and more closely associated tasks as well.
- Several instances showed that even MLMs failed due to inter-language interferences. These issues could potentially be addressed by integrating multilingual Named Entity Recognition (NER) (Vitiugin et al., 2024).
- The pretraining of VMLMs can be more language inclusive, i.e., it should contain better data representation for code-mixed setting (Zhang et al., 2023).

Ethics statement:

All the datasets used in this paper are either publicly available or gathered directly from corresponding author with a permission to use for research purpose. No new data collection or annotation was done as part of our work, and hence we aren't releasing any dataset. It is important to note that the paper may contain offensive, mockery or discriminatory language towards certain individuals or groups. We acknowledge this and want to clarify that we do not agree with or support these views in any way. We strictly adhere to the Google Translate API's Terms of Service[3] for generating the translated Hindi datasets.

Acknowledgment

The authors thank Rada Mihalcea and Diyi Yang for sharing their datasets.

References

Ibrahim Abu Farha, Silviu Vlad Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 task 6: iSarcasmEval, intended sarcasm detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 802–814, Seattle, United States. Association for Computational Linguistics.

Muhammad Abulaish, Ashraf Kamal, and Mohammed J Zaki. 2020. A survey of figurative language and its computational detection in online social networks. *ACM Transactions on the Web (TWEB)*, 14(1):1–52.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kaustubh Agarwal and Rhythm Narula. 2021. Humor generation and detection in code-mixed Hindi-English. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 1–6, Online. INCOMA Ltd.

Akshita Aggarwal, Anshul Wadhawan, Anshima Chaudhary, and Kavita Maurya. 2020. "did you really mean what you said?" : Sarcasm detection in Hindi-English code-mixed data using bilingual word embeddings. In *Proceedings of the Sixth Workshop on Noisy User-generated Text*

[3] https://developers.google.com/terms

(W-NUT 2020), pages 7–15, Online. Association for Computational Linguistics.

Vikram Ahuja. 2019. Computational analysis of humour. *International Institute of Information Technology Hyderabad, India*.

Sejal Aloria, Ishika Aggarwal, Niyati Baliyan, and Mohona Ghosh. 2023. Hilarious or hidden? detecting sarcasm in hinglish tweets using bert-gru. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.

Iqra Ameer, Necva Bölücü, Hua Xu, and Ali Al Bataineh. 2023. Findings of WASSA 2023 shared task: Multi-label and multi-class emotion classification on code-mixed text messages. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 587–595, Toronto, Canada. Association for Computational Linguistics.

Issa Annamoradnejad and Gohar Zoghi. 2024. Colbert: Using bert sentence embedding in parallel neural networks for computational humor. *Expert Systems with Applications*, 249:123685.

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and contextual embedding for sarcasm detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 225–243, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Alexander Baranov, Vladimir Kniazhevsky, and Pavel Braslavski. 2023. You told me that joke twice: A systematic investigation of transferability and robustness of humor detection models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13701–13715, Singapore. Association for Computational Linguistics.

Elisa Bassignana, Valerio Basile, Viviana Patti, et al. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *CEUR Workshop proceedings*, volume 2253, pages 1–6. CEUR-WS.

Manjot Bedi, Shivani Kumar, Md Shad Akhtar, and Tanmoy Chakraborty. 2023. Multi-modal sarcasm detection and humor classification in code-mixed conversations. *IEEE Transactions on Affective Computing*, 14(2):1363–1375.

Dario Bertero and Pascale Fung. 2016. A long short-term memory framework for predicting humor in dialogues. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 130–135, San Diego, California. Association for Computational Linguistics.

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.

Paul Bleakley and Daniel Sailofsky. 2023. Politics, jokes, and banter amid tragedy: the use of sarcasm and mocking on social media in response to the uvalde school shooting. *The Journal of Social Media in Society*, 12(2):62–81.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. 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.

Wangqun Chen, Fuqiang Lin, Guowei Li, and Bo Liu. 2024a. A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities. *Neurocomputing*, page 127428.

Yuyan Chen, Yichen Yuan, Panjun Liu, Dayiheng Liu, Qinghao Guan, Mengfei Guo, Haiming Peng, Bang Liu, Zhixu Li, and Yanghua Xiao. 2024b. Talk funny! a large-scale humor response dataset with chain-of-humor interpretation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17826–17834.

Luis Chiruzzo, Santiago Castro, Santiago Góngora, Aiala Rosa, J Meaney, and Rada Mihalcea. 2021. Overview of haha at iberlef 2021: Detecting, rating and analyzing humor in spanish. *Procesamiento de Lenguaje Natural*, 67:257–268.

Monojit Choudhury, Kalika Bali, Sunayana Sitaram, and Ashutosh Baheti. 2017. Curriculum design for code-switching: Experiments with language identification and language modeling with deep neural networks. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 65–74, Kolkata, India. NLP Association of India.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek,

Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online. Association for Computational Linguistics.

Mithun Das, Punyajoy Saha, Binny Mathew, and Animesh Mukherjee. 2022. HateCheckHIn: Evaluating Hindi hate speech detection models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5378–5387, Marseille, France. European Language Resources Association.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Suzanne Elayan, Martin Sykora, Thomas W Jackson, and Ejovwoke Onojeharho. 2022. 'are you having a laugh?': detecting humorous expressions on social media: an exploration of theory, current approaches and future work. *International Journal of Information Technology and Management*, 21(1):115–137.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Simona Frenda et al. 2018. The role of sarcasm in hate speech. a multilingual perspective. In *Proceedings of the Doctoral Symposium of the XXXIVInternational Conference of the Spanish Society for Natural Language Processing (SEPLN 2018)*, pages 13–17. Lloret, E.; Saquete, E.; Mart ınez-Barco, P.; Moreno, I.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California. Association for Computational Linguistics.

Fabrício Góes, Piotr Sawicki, Marek Grzes, Marco Volpe, and Daniel Brown. 2023. Is gpt-4 good enough to evaluate jokes? In *International Conference on Innovative Computing and Cloud Computing*.

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 Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. 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 2046–2056, Hong Kong, China. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. 2024. Evaluating code-switching translation with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguis-*

tics, Language Resources and Evaluation (LREC-COLING 2024), pages 6381–6394, Torino, Italia. ELRA and ICCL.

Suzana Ilić, Edison Marrese-Taylor, Jorge Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–7, Brussels, Belgium. Association for Computational Linguistics.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5):1–22.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, Mark Carman, Meghna Singh, Jaya Saraswati, and Rajita Shukla. 2016. How challenging is sarcasm versus irony classification?: A study with a dataset from English literature. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 123–127, Melbourne, Australia.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Ashraf Kamal and Muhammad Abulaish. 2020. Self-deprecating humor detection: A machine learning approach. In *Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16*, pages 483–494. Springer.

Mary Ogbuka Kenneth, Foaad Khosmood, and Abbas Edalat. 2024. Systematic literature review: Computational approaches for humour style classification. *arXiv preprint arXiv:2402.01759*.

Ankush Khandelwal, Sahil Swami, Syed S. Akhtar, and Manish Shrivastava. 2018. Humor detection in English-Hindi code-mixed social media content : Corpus and baseline system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Akshi Kumar, Abhishek Mallik, and Sanjay Kumar. 2023. Humourhindinet: Humour detection in hindi web series using word embedding and convolutional neural network. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Frances Adriana Laureano De Leon, Harish Tayyar Madabushi, and Mark Lee. 2024. Code-mixed probes show how pre-trained models generalise on code-switched text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3457–3468, Torino, Italia. ELRA and ICCL.

Da Li, Boqing Zhu, Sen Yang, Kele Xu, Ming Yi, Yukai He, and Huaimin Wang. 2023. Multi-task pre-training language model for semantic network completion. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(11).

Yang Liu and Yuexian Hou. 2023. Mining effective features using quantum entropy for humor recognition. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2048–2053, Dubrovnik, Croatia. Association for Computational Linguistics.

Yiyi Liu, Ruqing Zhang, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. 2023. Prompt tuning with contradictory intentions for sarcasm recognition. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–339, Dubrovnik, Croatia. Association for Computational Linguistics.

Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*,

pages 30–40, Atlanta, Georgia. Association for Computational Linguistics.

Krishanu Maity and Sriparna Saha. 2021. A multitask model for sentiment aided cyberbullying detection in code-mixed indian languages. In *Neural Information Processing*, pages 440–451, Cham. Springer International Publishing.

Debajyoti Mazumder, Aakash Kumar, and Jasabanta Patro. 2024. Improving code-mixed hate detection by native sample mixing: A case study for hindi-english code-mixed scenario. *arXiv preprint arXiv:2405.20755*.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Rada Mihalcea and Carlo Strapparava. 2005. Making computers laugh: Investigations in automatic humor recognition. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 531–538, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Abhijit Mishra, Tarun Tater, and Karthik Sankaranarayanan. 2019. A modular architecture for unsupervised sarcasm generation. 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 6144–6154, Hong Kong, China. Association for Computational Linguistics.

Rishabh Misra. 2022. News category dataset.

Rishabh Misra and Prahal Arora. 2019. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.

Chakita Muttaraju, Aakansha Singh, Anusha Kabber, and Mamatha H. R. 2022. Semi-supervised and unsupervised detection of humour in code-mixed Hindi-English tweets. In *Proceedings of the Third International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2022) co-located with ICNLSP 2022*, pages 8–13, Trento, Italy. Association for Computational Linguistics.

Sripriya N, Thenmozhi Durairaj, Nandhini K, Bharathi B, Kishore Kumar Ponnusamy, Charmathi Rajkumar, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Subalalitha Chinnaudayar Navaneethakrishnan, and Bharathi Raja Chakravarthi. 2024. Findings of shared task on sarcasm identification in code-mixed dravidian languages. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '23, page 22–24, New York, NY, USA. Association for Computing Machinery.

Arijit Nag, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2024. Cost-performance optimization for processing low-resource language tasks using commercial LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15681–15701, Miami, Florida, USA. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles. Association for Computational Linguistics.

Rajnish Pandey and Jyoti Prakash Singh. 2023. Bert-lstm model for sarcasm detection in code-mixed social media post. *Journal of Intelligent Information Systems*, 60(1):235–254.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1791–1799, Seattle, United States. Association for Computational Linguistics.

Jasabanta Patro, Srijan Bansal, and Animesh Mukherjee. 2019. A deep-learning framework to detect sarcasm targets. 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 6336–6342, Hong Kong, China. Association for Computational Linguistics.

Jasabanta Patro, Bidisha Samanta, Saurabh Singh, Abhipsa Basu, Prithwish Mukherjee, Monojit Choudhury, and Animesh Mukherjee. 2017. All that is english may be hindi: Enhancing language identification through automatic ranking of the likeliness of word borrowing in social media. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2264–2274.

Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: Interpreting sarcasm with sentiment based

monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1690–1700, Vancouver, Canada. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Amruta Purandare and Diane Litman. 2006. Humor: Prosody analysis and automatic recognition for F*R*I*E*N*D*S*. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 208–215, Sydney, Australia. Association for Computational Linguistics.

Anil Ramakrishna, Timothy Greer, David Atkins, and Shrikanth Narayanan. 2018. Computational Modeling of Conversational Humor in Psychotherapy. In *Proc. Interspeech 2018*, pages 2344–2348.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Guy Rotman and Roi Reichart. 2022. Multi-task active learning for pre-trained transformer-based models. *Transactions of the Association for Computational Linguistics*, 10:1209–1228.

Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika Mamidi. 2019a. Deep learning techniques for humor detection in Hindi-English code-mixed tweets. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 57–61, Minneapolis, USA. Association for Computational Linguistics.

Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika Mamidi. 2019b. Stance detection in code-mixed Hindi-English social media data using multi-task learning. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 1–5, Minneapolis, USA. Association for Computational Linguistics.

Aditya Shah and Chandresh Maurya. 2021. How effective is incongruity? implications for code-mixed sarcasm detection. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 271–276, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Aditya Shah and Chandresh Kumar Maurya. 2022. How effective is incongruity? implications for code-mix sarcasm detection. *arXiv preprint arXiv:2202.02702*.

Wenbo Shang, Jiangjiang Zhao, Zezhong Wang, Binyang Li, Fangchun Yang, and Kam-Fai Wong. 2022. "I know who you are": Character-based features for conversational humor recognition in Chinese. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2927–2932, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vaibhav Shukla, Manjira Sinha, and Tirthankar Dasgupta. 2019. Automatic humor detection from code-mixed tweets. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '19, page 56–59, New York, NY, USA. Association for Computing Machinery.

Bhuvanesh Singh and Dilip Kumar Sharma. 2023. A survey of sarcasm detection techniques in natural language processing. In *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–6. IEEE.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Oliviero Stock and Carlo Strapparava. 2005. HAHAcronym: A computational humor system. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 113–116, Ann Arbor, Michigan. Association for Computational Linguistics.

Rahmawati Sukmaningrum. 2018. The analysis of translation techniques of irony and sarcasm in novel entitled the sign of the four. *ETERNAL (English Teaching Journal)*, 7.

Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

Leonard Tang, Alexander Cai, Steve Li, and Jason Wang. 2022. The naughtyformer: A transformer understands offensive humor.

Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. MVP: Multi-task supervised pre-training for natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8758–8794, Toronto, Canada. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Joseph Tepperman, David Traum, and Shrikanth Narayanan. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm — a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. *Proceedings of the International AAAI Conference on Web and Social Media*, 4(1):162–169.

G Richard Tucker. 1999. A global perspective on bilingualism and bilingual education. *Washington, DC*.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset for detecting irony in hindi-english code-mixed social media text. In *Proceedings of 4th Workshop on Sentic Computing, Sentiment Analysis, Opinion Mining, and Emotion Detection (EMSASW 2018) Co-located with the 15th Extended Semantic Web Conference 2018 (ESWC 2018), Heraklion, Greece, June 4, 2018*, volume 2111 of *CEUR Workshop Proceedings*, pages 38–46. CEUR-WS.org.

Fedor Vitiugin, Sunok Lee, Henna Paakki, Anastasiia Chizhikova, and Nitin Sawhney. 2024. Unraveling code-mixing patterns in migration discourse: Automated detection and analysis of online conversations on reddit. *arXiv preprint arXiv:2406.08633*.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817, Istanbul, Turkey. European Language Resources Association (ELRA).

Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. PARIKSHA: A large-scale investigation of human-LLM evaluator agreement on multilingual and multi-cultural data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7900–7932, Miami, Florida, USA. Association for Computational Linguistics.

Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. 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 3621–3625, Hong Kong, China. Association for Computational Linguistics.

Orion Weller and Kevin Seppi. 2020. The rJokes dataset: a large scale humor collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6136–6141, Marseille, France. European Language Resources Association.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada. Association for Computational Linguistics.

Qi Wu, Peng Wang, and Chenghao Huang. 2020. MeisterMorxrc at SemEval-2020 task 9: Fine-tune bert and multitask learning for sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1294–1297, Barcelona (online). International Committee for Computational Linguistics.

Yukang Xie, Chengyu Wang, Junbing Yan, Jiyong Zhou, Feiqi Deng, and Jun Huang. 2024. Making small language models better multi-task learners with mixture-of-task-adapters. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 1094–1097, New York, NY, USA. Association for Computing Machinery.

Sargam Yadav, Abhishek Kaushik, and Kevin McDaid. 2024. Leveraging weakly annotated data for hate speech detection in code-mixed hinglish: A feasibility-driven transfer learning approach with large language models. *arXiv preprint arXiv:2403.02121*.

Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2367–2376, Lisbon, Portugal. Association for Computational Linguistics.

Zixiaofan Yang, Shayan Hooshmand, and Julia Hirschberg. 2021. CHoRaL: Collecting humor reaction labels from millions of social media users. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4429–4435, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan. The COLING 2016 Organizing Committee.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri Aji. 2023. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. Sarcasmbench: Towards evaluating large language models on sarcasm understanding. *arXiv preprint arXiv:2408.11319*.

Wenye Zhao, Qingbao Huang, Dongsheng Xu, and Peizhi Zhao. 2023. Multi-modal sarcasm generation: Dataset and solution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5601–5613, Toronto, Canada. Association for Computational Linguistics.

Zhenjie Zhao, Andrew Cattle, Evangelos Papalexakis, and Xiaojuan Ma. 2019. Embedding lexical features via tensor decomposition for small sample humor recognition. 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 6376–6381, Hong Kong, China. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Related works:

In this section, we discussed the past works of this domain. We listed task-wise description of the past literature in the following subsections.

A.1 Monolingual humor:

The automatic detection of humor has gathered significant interest in NLP community. Most of the research in past literature focused on monolingual setting. From the task point of view, we saw binary classification ('humor' or 'non-humor') (Mihalcea and Strapparava, 2005; Purandare and Litman, 2006; Yang et al., 2015; Ramakrishna et al., 2018; Hasan et al., 2019; Zhao et al., 2019; Liu and Hou, 2023), multi-class and multi-label classification on humor targets (Chiruzzo et al., 2021), ranking (top 10 humorous utterances) (Zhao et al., 2019), scoring (based on a reaction based humor score) (Yang et al., 2021), generation (Stock and Strapparava, 2005; Chen et al., 2024b), etc. From the approach point of view, past studies explored various approaches, ranging from statistical techniques to deep learning models (Abulaish et al., 2020). For instance, initially researchers looked into stylistic features such as alliteration, antonyms, and adult slangs (Mihalcea and Strapparava, 2005), and also prosody features like pitch, tempo, and emphasis (Purandare and Litman, 2006). In addition, (Stock and Strapparava, 2005) used theoretical ideas like incongruity theory to generate humorous acronyms. Later, with the rise of deep-neural networks like RNNs, LSTMs, and CNNs with character ngram, Word2Vec and kNN features(Yang et al., 2015; Bertero and Fung, 2016; Ramakrishna et al., 2018; Hasan et al., 2019)), research on computational humor identification has grown significantly. The development of attention mechanisms allowed identification context-based humor using transformer models (Weller and Seppi, 2019) and pretrained language models (Yang et al., 2021; Shang et al., 2022; Chen et al., 2024b).

A.2 Code-mixed humor:

Very few studies focused on humor detection in code-mixed environments, and most of them

are present in the Indian context (Khandelwal et al., 2018; Sane et al., 2019a; Shukla et al., 2019; Bedi et al., 2023). Khandelwal et al. (2018) introduced the first Hindi-English code-mixed dataset and evaluated it on statistical classifiers (like SVM, NB and Random Forest) with n-gram and bag-of-words features for humor detection in code-mixed setting. This dataset served as a valuable resource for subsequent studies in the field. Sane et al. (2019a) proposed an attention-based bidirectional LSTM model using Continuous Bag of Words (CBOW) and Skip-gram embeddings for humor detection in the same code-mixed dataset. Building upon existing research, Shukla et al. (2019) compared bidirectional LSTM and CNN models utilizing word and sentence embeddings for humor detection in Hindi-English code-mixed text. Their study provided insights into the comparative efficacy of different neural network architectures for this task. In a recent development, Bedi et al. (2023) leveraged contextual attention mechanisms for multi-modal humor detection in Hindi-English code-mixed text. This approach highlighted the importance of considering contextual cues in code-mixed humor analysis.

A.3 Monolingual sarcasm:

Sarcasm detection, though a challenging task, has gained attention due to its repurcussions. It is also considered as an implicit hate. From the task point of view, we saw many variations, like binary classification ('sarcastic' or 'non-sarcastic') (Tepperman et al., 2006; Tsur et al., 2010; Joshi et al., 2015; Riloff et al., 2013), sarcasm generation (Mishra et al., 2019; Zhao et al., 2023; Ilić et al., 2018), counter sarcasm generation (Peled and Reichart, 2017), translation (Sukmaningrum, 2018), etc. From an architectural point of view, researchers explored rule-based features such as prosodic, spectral, and contextual cues, along with patterns and punctuations (Tepperman et al., 2006), pattern matching by extracting High-Frequency Words (HFWs) and Content Words (CWs) (Tsur et al., 2010) and shift of sentiment and various incongruities (Joshi et al., 2015). Later on, deep learning-based approaches gained pace and demonstrated promising results. The RNN, LSTM and CNN (or a combination of them) networks claimed to show strengths of semantic modelling (Ghosh and Veale, 2016; Zhang et al., 2016). Further, the combination of contextual pretrained embeddings and socio-linguistic features such as Named Entity Recognition (NER), Part-of-Speech (POS) tagging, Empath, and LIWC (Linguistic Inquiry and Word Count) features proved to be better at sarcasm classification (Patro et al., 2019). Later, the attention mechanism based pretrained transformers came into play to capture better contextual sarcastic cues from the sequence of text (Potamias et al., 2020; Babanejad et al., 2020).

A.4 Code-mixed sarcasm:

Lately, the focus on code-mixed sarcasm detection started to come up (N et al., 2024; Bedi et al., 2023; Aggarwal et al., 2020; Shah and Maurya, 2021). Still the studies are limited in terms of datasets and architecture both. From the dataset point of view, we saw code-mixed sarcasm in Tamil-English and Malayalam-English (N et al., 2024), Hindi-English (Swami et al., 2018; Aggarwal et al., 2020; Vijay et al., 2018), multimodal Hindi-English (Bedi et al., 2023), etc. However, not all of the datasets are publicly available. From the methodological point, Aggarwal et al. (2020) uncovered a thorough comparison of deep learning based models like CNNs, LSTMs and Bi-directional LSTMs. Finally, the advent of transformers allowed researchers to utilize sub-word level embeddings (Shah and Maurya, 2021) and contextual attention mechanism for multi-modal Hindi-English code-mixed sarcasm detection (Bedi et al., 2023).

A.5 Multi-task learning:

Multi-task learning has been used widely in the field of natural language processing for the past few years (Xie et al., 2024; Li et al., 2023; Tang et al., 2023). From the task point of view, particularly in the code-mixed scenario, it has shown some promising results in various tasks such as stance detection (Sane et al., 2019b), sentiment analysis (Wu et al., 2020), cyberbullying detection (Maity and Saha, 2021), and emotion classification (Ameer et al., 2023). Architecture-wise, researchers utilized multi-channel CNN (Sane et al., 2019b), pretrained BERT model (Wu et al., 2020), BERT+VecMap (Maity and Saha, 2021), pretrained LMs and prompt tuning (Ameer et al.,

2023). However, the quantity of data has minimal impact on these tasks; instead, utilizing data from diverse sources has proven to be a more effective solution (Baranov et al., 2023). Finetuning MLMs with task specific modules (e.g. adapters) achieved success in cross-lingual learning for low-resource languages (Pfeiffer et al., 2020; Parović et al., 2022).

A.6 In-context learning:

Although the very large language models are known for their knowledge and ability to perform well with just a few examples, their comprehension of low resource languages is still suboptimal (Bang et al., 2023). From the prompting point of view, Liu et al. (2023) proposed prompt tuning, where they defined several prompt templates and verbalizers to assess whether the intended meaning of a comment contradicts the content provided in the prompt. Past work (Nag et al., 2024) have compared the performance of native script and translated/ transliterated version of it for various tasks like sentiment classification, paraphrasing, intent classification, summarization, question answering, multichoice question answering, etc.

A.7 Research gap:

Laureano De Leon et al. (2024) reported that pretrained MLMs retained enough native language information for processing code-mixed text containing closely associated languages like Spanish and English. Thus, the MLMs improved in sentiment related tasks like code-mixed hate detection via native sample mixing (Mazumder et al., 2024), but require native samples from the participating languages. Choudhury et al. (2017) also showed that a generic DNN performs best in LID and language modeling task when native samples are either mixed with code-mixed data or trained sequentially- first with native samples, followed by code-mixed data. As a result, past studies highlighted the benefits of native sample mixing and multi-task learning in various tasks. To the best of our knowledge, there is no existing study which uses both native samples and multitasking for code-mixed humor and sarcasm detection.

B Code-mixing:

Code-mixing is a linguistic phenomenon common across multilingual speakers. Multilingual speakers are believed to outnumber monolingual speakers globally (Tucker, 1999). A large chunck of population in Asia, Europe, and North America know more then one language, this allows them to communicate among themselves by switching languages in a single utterance. It is more prominent in informal communications like social media posts and voice mails (Patro et al., 2017).

C Additional dataset details:

In this section, we described the details of code-mixed and monolingual datasets considered in our study. They were arranged task-wise in the following subsections.

C.1 Humour:

We could find only one publicly available Hindi-English code-mixed humour dataset, introduced by **Khandelwal et al. (2018)** (GPL 3.0 licensed). However, there are several English humour datasets proposed in the literature (Annamoradnejad and Zoghi, 2024; Yang et al., 2015; Meaney et al., 2021; Mihalcea and Strapparava, 2005; Weller and Seppi, 2020; Tang et al., 2022; Kamal and Abulaish, 2020). Out of which, we considered ColBERT (**Col**) (Annamoradnejad and Zoghi, 2024), Pun of the Day (**POTD**) (Yang et al., 2015), HaHackathon (**HaHa**) (Meaney et al., 2021) and 16000 One Liners (**16000**) (Mihalcea and Strapparava, 2005) datasets. This is because they are relatively balanced and widely used in past studies(Kenneth et al., 2024). On the contrary, we couldn't find any appropriate Hindi humour dataset. Although Kumar et al. (2023) recently proposed one, it comprised dialogues taken from a famous Hindi TV series. Further, upon deep inspection, we found that the associated labels depend highly on preceding dialogues with varying contexts (dialogue iterations). Thus, we restricted ourselves from using them. In the following, we report a brief description of the individual datasets,

C.1.1 Code-mixed dataset:

- **Khandelwal et al. (2018):** Khandelwal et al. (2018) gathered 10,478 tweets

	Language	Dataset	Class	Sample
Humor	Code-mixed	Khandelwal et al. (2018)	$+ve$	Difference between Sidhu and Amir Khan? Sidhu TV par hansne ke paise lete hain aur Amir Khan rone ke. (**Translation:** They might be culturally rich, but my country is also a haven for the uneducated. #SuSaid #irony #india)
			$-ve$	Match dekhna start kiya to ABD out ho gaya. (**Translation:** Started watching the match and ABD got out.)
	English	ColBERT	$+ve$	"What's the tallest building in your city? the library, because it has the most stories." #murderer
			$-ve$	Meet the brilliant pianist behind martin scorsese's upcoming biopic
	English	Pun of the Day	$+ve$	my new theory on inertia doesn t seem to be gaining momentum
			$-ve$	God could not be everywhere and therefore he made mothers
	English	Hahackathon	$+ve$	What happens if Usain Bolt misses his bus? He waits for it at the next stop.
			$-ve$	"Forgiving is easy, it's trusting again which is the hard part"
	English	16000 One Liners	$+ve$	"If going to church makes you a Christian, does going to a garage make you a car?"
			$-ve$	But now I'm fulfilled. . . SO MAKE ME A SANDWICH!!!
Sarcasm	Code-mixed	Swami et al. (1805)	$+ve$	Culturally rich honge..par gavaaro ki basti bhi mera desh. #SuSaid #irony #india (**Translation:** They might be culturally rich, but my country is also a haven for the uneducated. #SuSaid #irony #india)
			$-ve$	Yeh kya vahiyaat kisam ki ODI cricket hai? #SLvIND (**Translation:** What kind of ridiculous ODI cricket is this? #SLvIND)
	English	News Headlines Dataset	$+ve$	taliban agrees to peace deal despite concerns about america's human-rights record
			$-ve$	this dancing traffic light is the grooviest way for pedestrians to stay safe
	English	iSarcasm	$+ve$	did you know there is a direct correlation with how cool someone is and how high they cuff their pants?
			$-ve$	why are salads so expensive?!It's like I'm being punished for tryna eat healthy...
	English	SC-V2	$+ve$	what about God? So many God-fearing people, he must be a terrorist :)
			$-ve$	"Hmm, while she is right at the first point, the second doesn't follow."
Hate	Code-mixed	Bohra et al. (2018)	$+ve$	4G bolkar 2G ka speed mil raha hai ye sara sar dokha hai. I hate jio (**Translation:** Claiming 4G but getting the speed of 2G is a complete deception. I hate Jio.)
			$-ve$	ye vidya salman se bhut bdi celebrity h. (**Translation:** This Vidya is a much bigger celebrity than Salman.)
	Hindi	HateCheckHIn	$+ve$	मेरे में इन अंग्रेज़ों के लिए नरफत के अलावा कुछ नहीं है (**Translation:** I have nothing but hatred for these English people.)
			$-ve$	मैं कभी किसी महिला को चोट नहीं पहुंचाऊंगा (**Translation:** i would never hurt a woman)
	English	HASOC-2019	$+ve$	@GovPritzker You are an evil man. #murderer
			$-ve$	E. Jean Carroll responds to Trump saying she's 'not his type

Figure 1: Samples from all the datasets. The Hindi parts (Romanized or Devanagari) were underlined for identification. We have provided the English translations of code-mixed and Hindi samples for readability.

from various domains like 'sports', 'entertainment', 'politics', etc. They manually identified 3,453 code-mixed tweets by discarding monolingual Hindi and English tweets. Each identified tweet was annotated by three language experts, skilled in both Hindi and English. They tagged each tweet as "humorous" (H) or "non-humorous" (N). As a norm, they labelled the tweets containing anecdotes, fantasy, irony, jokes, and insults as humorous, and the tweets with facts, dialogues and speeches that didn't provoke any laughter were labelled non-humorous. We found that, over time, some of the tweets in the original dataset got deleted. Therefore, we resorted to using the currently available samples, which is a total of 2,951 tweets (1,759 humorous and 1,192 non-humorous).

C.1.2 Native language datasets:

- **ColBERT (Col)(Annamoradnejad and Zoghi, 2024):** ColBERT (Annamoradnejad and Zoghi, 2024) was formed by combining samples from two previously published datasets: (i) news website (Misra, 2022) and (ii) jokes website (Weller and Seppi, 2019). The news website dataset consists of 200k Huffington Post news headlines from 2012-2018. They are from different categories like politics, wellness, entertainment and parenting. The jokes website dataset consists of around 231k humorous samples collected from two subreddits : /r/jokes and /r/cleanjokes. Their authors randomly selected 100k samples from both datasets after a few fine-grained preprocessing steps including de-duplication of samples, lexical statistics matching and title case formatting.

- **Pun of the Day (POTD)(Yang et al., 2015):** The Pun of the Day (Yang et al., 2015) (MIT licensed) dataset consists of humorous samples directly collected from the pun-of-the-day jokes website[4]. The non-humorous samples were scraped from AP news, the New York Times, Yahoo! Answer and Proverb websites. Their authors performed a curated sampling of negative samples to minimize domain differences.

- **HaHackathon (HaHa)(Meaney et al., 2021):** The HaHackathon (Meaney et al., 2021) comprised of 10k samples. Meaney et al. (2021) created this dataset with Twitter posts (80%) and Kaggle Short Jokes samples[5] (20%). While the humorous tweets were collected from humorous Twitter accounts (e.g. @humurous1liners and @conanobrien), and the non-humorous tweets were collected from some celebrity accounts (e.g. @thatonequeen and @Oprah). From the Kaggle dataset, they selected samples expressing humour and offence. The accumulated 10k samples were annotated by twenty US-based annotators aged between 18 and 70 years, by answering the question *"Is the intention of this text to be humorous?"*.

- **16000 One Liners (16000)(Mihalcea and Strapparava, 2005):** This dataset (Mihalcea and Strapparava, 2005) consists of 32k short sentences. Out of which, 16k are humorous samples. They were automatically collected through a web-based bootstrapping process. The remaining 16k samples are non-humorous, and they were collected from Reuters titles, Proverbs and British National Corpus (BNC).

C.2 Sarcasm:

We could find two publicly available Hindi-English code-mixed sarcasm datasets. They were introduced by **Swami et al. (2018)** and Shah and Maurya (2022). Out of them, Shah and Maurya (2022) distantly labelled their samples with the help of hashtags associated with tweets. We restricted ourselves from using it as our primary objective is to improve code-mixed sarcasm and introducing this dataset can make the results noisy. There are several native English sarcasm datasets present as well (Misra and Arora, 2019; Joshi et al., 2016; Abu Farha et al., 2022; Ptáček et al., 2014; Oprea and Magdy, 2020; Lukin and Walker, 2013; Oraby et al., 2016). In the present work, we considered to experiment with News Headlines Dataset

[4] http://www.punoftheday.com

[5] https://www.kaggle.com/abhinavmoudgil95/short-jokes

(**NHD**) (Misra and Arora, 2019), iSarcasm (**iSarc**) (Abu Farha et al., 2022) and Sarcasm Corpus V2 (**SC-V2** hence after) (Oraby et al., 2016); as they were widely studied in literature(Joshi et al., 2017; Chen et al., 2024a). iSarcasm and SC-V2 were manually annotated by expert annotators, whereas the News Headlines Dataset is a distant labelled dataset. Similar to the case of Hindi humour detection, we could not find any publicly available Hindi sarcasm dataset. In the following, we provided a brief description of the considered sarcasm datasets:

C.2.1 Code-mixed dataset:

- **Swami et al. (2018):** To create this dataset (GPL-3.0 licensed), Swami et al. (2018) scrapped tweets containing keywords 'politics', 'cricket', and 'Bollywood'. They manually filtered Hindi-English code-mixed tweets by inspecting individual samples. Tweets containing '#sarcasm' and '#irony' and lacking them are kept in the initial pool of sarcastic and non-sarcastic samples, respectively. A group of language experts well-versed in Hindi and English annotated the samples with an inter-annotator agreement (Fleiss and Cohen, 1973) of 0.79. The final version of their dataset has 5,250 samples, out of which 504 are sarcastic.

C.2.2 Native language datasets:

- **News Headlines Dataset (NHD)(Misra and Arora, 2019):** The samples collected in this dataset were headlines from two websites: (i) TheOnion[6] briefs, comprises of sarcastic explanations of current events (as sarcastic samples) and (ii) HuffPost[7], an American news website (as non-sarcastic samples). They down-sampled HuffPost samples to nearly match the sarcastic samples, resulting in a balanced dataset of nearly 26.7k samples.

- **iSarcasm (iSarc)(Abu Farha et al., 2022):** This dataset consists of 5,735 tweets implicitly labelled by tweet authors. Abu Farha et al. (2022) conducted a survey among English speakers having Twitter accounts. Participants were asked to provide one link to their sarcastic and three links to their non-sarcastic tweets posted in their recent past. Additionally, the authors also requested the survey participants to provide a non-sarcastic version of their sarcastic tweets.

- **SC-V2(Oraby et al., 2016):** This dataset consists of 9,386 text samples collected from three different online debate forums like `4forums.com`, `CreateDebate.com` and `Convinceme.Net`. Nine expert annotators then annotated each sample as 'sarcastic' or 'not-sarcastic'. Further, annotators also labelled them for three subtypes of sarcasm: general, hyperbole and rhetorical questions. This dataset is a subset of the Internet Argument Corpus (IAC) (Walker et al., 2012).

C.3 Hate:

To facilitate knowledge sharing across tasks in MTL frameworks, we used publicly available Hindi-English code-mixed and native (i.e. monolingual Hindi and English) hate datasets. We could find only one Hindi Das et al. (2022) (hereafter referred to as **HCHIn**) and Hindi-English code-mixed **Bohra et al. (2018)** hate dataset that is publicly available. Further, as an English hate dataset, we used HASOC-2019 (English) (**HASOC** hence after)[8] as it is widely used in past works. In the following, we provide a brief description of the individual datasets:

C.3.1 Code-mixed dataset:

- **Bohra et al. (2018):** To create this dataset, authors retrieved 112,718 tweets based on a predefined list of hashtags and keywords related to 'politics', 'public protests', 'riots', etc. Following this, 4,575 code-mixed tweets were manually filtered, and two expert annotators tagged them as "hate"(H) or "non-hate"(NH).

C.3.2 Native language datasets:

- **HateCheckHIn (HCIn)(Das et al., 2022):** This dataset contains 4,754 Hindi samples, each annotated with 'hate' or 'non-hate' by expert annotators well-versed in the Hindi language. This dataset was

[6] `https://www.theonion.com/`
[7] `https://www.huffpost.com/`
[8] HASOC-2019: `https://hasocfire.github.io/hasoc/2019/dataset.html`

constructed to test the weaknesses of Hindi hate speech detection models. Das et al. (2022) manually designed 28 monolingual functionality tests for that purpose. The quality of the test cases was verified by two expert annotators.

- **HASOC-2019[8] (HASOC):** It contains 5,852 social media posts collected from Twitter and Facebook using hashtags and keywords. Following this, each sample was annotated as 'hate' or 'non-hate' by organizers of the HASOC track.

D Additional baseline details:

In this section, we described the details of baseline methods considered in our study. They were arranged task-wise in the following subsections.

D.1 Humor:

- Agarwal and Narula (2021) experimented with various neural architectures ranging from variants of LSTMs (such as vanilla LSTM, Bi-LSTM and Bi-LSTM with attention mechanism) to MLMs like mBERT (Devlin et al., 2019) and IndicBERT (Kakwani et al., 2020). They found that MLMs by far outperform the LSTMs in terms of accuracy. IndicBERT is pre-trained on 12 major Indian languages (which includes 1.84 B Hindi tokens) with fewer parameters than mBERT (Devlin et al., 2019). We considered both language models as baseline.

- Muttaraju et al. (2022) approached the problem in a semi-supervised manner. They used a ratio of 1:100 for labeled versus unlabeled data. The labeled subset was used to train a classifier at first, and then they utilized the same classifier to get pseudo-labels from the unlabeled data points based on a threshold of prediction probability. The new training set for supervised modeling now consists of both pseudo-labeled and gold-labeled samples. This process is repeated until either the maximum number of iterations is reached or no more labeled data remains. From modeling point of view, they utilized HinglishBERT[9] within the GAN-BERT(Croce et al., 2020) architecture.

D.2 Sarcasm:

- Pandey and Singh (2023) experimented on a variety of neural architectures ranging from linear layers, CNNs, and LSTMs to pre-trained BERT(Devlin et al., 2019) and BERT-LSTM (LSTM stacked upon mBERT). LSTM-BERT significantly outperformed the others in terms of F1 score of positive class. Thus, we considered it as our baseline.

- Aloria et al. (2023) preprocessed the individual samples using a spelling-checker[10]. They experimented with a variety of architectures like CNNs, bi-LSTMs, statistical ensemble classifiers, BERT-LSTM and a novel BERT-GRU (bi-directional GRU stacked upon mBERT) architecture. They found that BERT-GRU significantly outperforms others in terms of the F1 score of the positive class.

E Additional details of Exp. 3:

E.1 Prompting details:

In this section, we described the prompting details for in-context learning for conducting **Exp. 3** (refer Section 3.4). Our prompt consisted of three parts, i) system prompt, where we explained the task, ii) few shots, examples that we fed to learn from, and iii) user input, which is the query for which we needed the predicted label. We selected the few shots through clustering technique which is considered to be a better approach (Huzaifah et al., 2024) than randomly picking examples. We presented the prompt template in Figure 2. The detailed results for the first scenario of few-shot prompting VMLM (Experiment-3) is presented in Table-7. The VMLMs were prompted with 0-shot and k-shot examples given in the context. The user input query remained in code-mixed language in each of the cases.

[9] https://huggingface.co/nirantk/hinglish-bert

[10] https://pypi.org/project/pyspellchecker/

System prompt

"""You are a humor recognition assistant judging if an 'Input' is humorous or not.
If the 'Input' is humorous, you need to give your final 'Output' as 'Humor'.
If the 'Input' is non-humorous, you need to give your final 'Output' as 'Non-humor'."""

Few shots

1 Input : 'Bhaiya 5 rupye ka sense dena.'
Output : 'Humor'

2 Input : 'Game is too fast yaar kya chal raha hai'
Output : 'Non-humor'

⋮

k Input : 'Kejriwal has a point. Lootne ka adhikar Aam Admi ko bhi chahiye..'
Output : 'Humor'

User Input

Input : 'Akmal bhai aaj butter toast khaya tha kya??'
Output :

Figure 2: Prompt template for k-shot prompting utilized for humor detection. English parts are marked in red and the Hindi parts are marked in blue.

E.2 LoRA-adapter based finetuning:

For the second scenario, we performed instruction finetuning on the VMLMs using the default parameters of LoRA adapter based supervised finetuning given in LLaMA-Factory (Zheng et al., 2024). We utilized the same prompt template for giving instruction to the models.

E.3 Detailed observations from Exp. 3:

In this section, we reported the additional observations of third experiment. While investigating the poor performance of the VMLMs, we identified certain patterns. The VMLMs were prompted to provide reasoning for their label predictions. Here are some observations,

- VMLMs favored specific labels like humour and sarcasm, where it extracted comedic or ironic effect in plain non-humourous statements. For instance, in the non-humourous sample *'So jao sab, kal Monday hai.'* (**Gloss:** 'Go to sleep, everyone. Tomorrow is Monday.'), Llama-3.1 predicted it as humourous and stated the reason : *'The input is a Hindi phrase that translates to "So go everyone, it's Monday." The humour lies in the fact that it's a common expression that people use to bid farewell on Fridays, but it's being used on Monday, which is the opposite day. It's a play on expectations and a lighthearted way to poke fun at the monotony of the workweek.'*.

- We could find instances where VMLM assumed some prior context over the original non-sarcastic sample *'Kabhi triple talaq halala pe v kuch post kardo agar himmat hai to'* (**Gloss:** 'Try posting something about triple talaq and halala if you have the courage.'), GPT-4 predicted it as sarcastic and stated the reason as *'The statement is sarcastic as it challenges the recipient's courage ("agar himmat hai to"), implying reluctance or hypocrisy. It uses a mocking tone to suggest selective bias in*

Humor

Model	Dataset	0-shot	2-shot	4-shot	8-shot	12-shot
Gemma	CM	**0.09**	0.08	0.08	**0.09**	0.07
	Col		0.08	**0.09**	0.04	0.04
	POTD		0.04	0.03	**0.07**	**0.07**
	HaHa		0.10	**0.11**	0.02	0.10
	16000		**0.10**	0.02	0.04	0.02
Aya Expanse	CM	0.72	0.72	0.72	**0.73**	0.72
	Col		**0.74**	0.71	0.73	0.69
	POTD		0.70	**0.73**	0.72	0.68
	HaHa		**0.71**	0.68	0.67	0.63
	16000		0.72	**0.73**	**0.73**	0.72
Llama-3.1	CM	**0.75**	**0.75**	0.74	**0.75**	**0.75**
	Col		0.72	**0.74**	0.68	0.63
	POTD		0.75	**0.76**	0.73	0.65
	HaHa		**0.75**	**0.75**	0.55	0.62
	16000		0.67	**0.77**	0.64	0.63
GPT-4	CM	0.73	0.73	**0.74**	**0.74**	0.73
	Col		0.47	0.51	0.56	**0.57**
	POTD		0.50	0.55	**0.62**	0.56
	HaHa		0.55	0.50	**0.56**	0.55
	16000		0.40	0.38	0.47	**0.58**

Sarcasm

Model	Dataset	0-shot	2-shot	4-shot	8-shot	12-shot
Gemma	CM	0.30	0.29	0.30	**0.34**	0.21
	NHD		**0.11**	**0.11**	**0.11**	0.02
	iSarc		0.35	**0.39**	0.37	0.30
	SC-V2		0.18	**0.47**	**0.47**	0.18
Aya Expanse	CM	0.18	**0.21**	0.20	**0.21**	0.18
	NHD		**0.21**	0.20	**0.21**	0.20
	iSarc		0.21	0.21	**0.26**	0.21
	SC-V2		0.21	0.21	**0.23**	0.21
Llama-3.1	CM	0.18	0.18	0.18	**0.21**	0.18
	NHD		0.30	**0.45**	0.37	0.36
	iSarc		0.27	0.31	0.32	**0.51**
	SC-V2		0.24	0.26	0.22	**0.28**
GPT-4	CM	0.76	0.77	0.76	**0.78**	0.75
	NHD		0.12	0.16	0.16	**0.17**
	iSarc		0.20	0.23	**0.25**	0.24
	SC-V2		0.40	**0.43**	0.42	0.41

Table 7: Results of prompting VMLMs with in-context few shot examples. Here, 0-shot represents providing no example in the context of the prompt, and k-shot represents k examples of input-output pairs are given in the context of the prompt. Notation: CM for code-mixed.

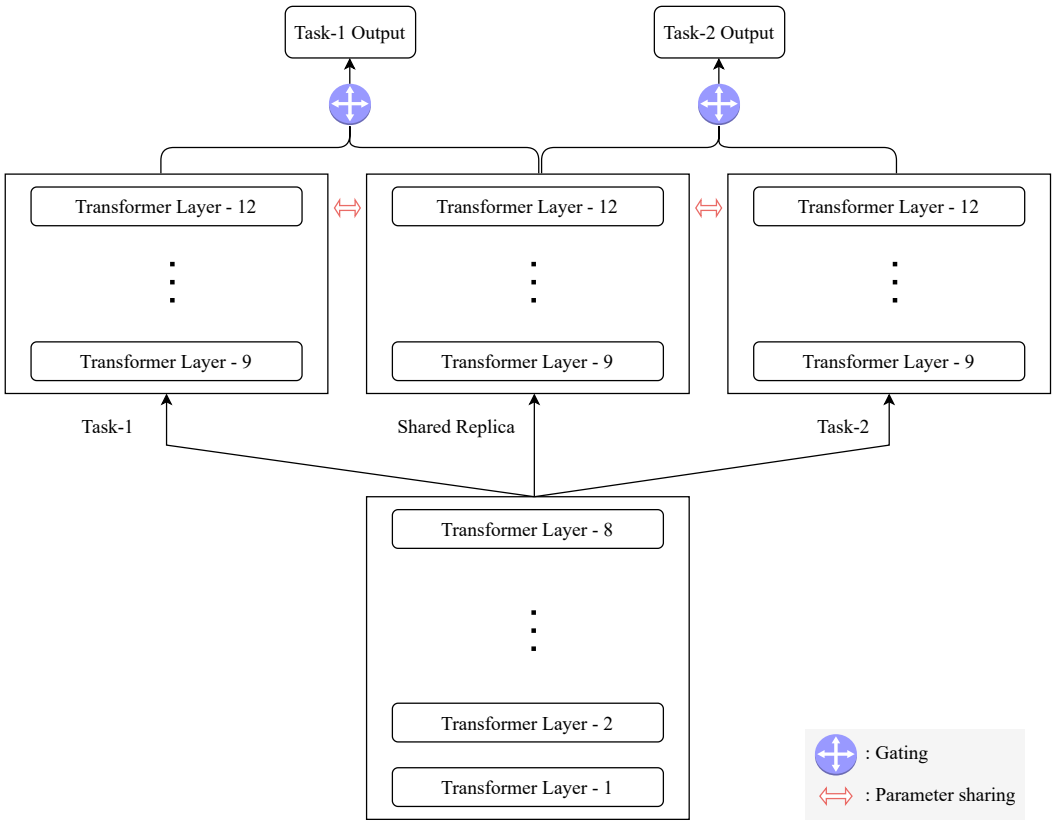


Figure 3: MTL architecture when number of tasks is two, i.e., $T = 2$.

posting about issues. The indirect criticism and taunting nature make it a clear case of sarcasm.'

- VMLMs showed a tendency to pick keywords, like in the non-sarcastic example *'Yeh bhee cricket kee deewani hain....'* (**Gloss:** 'She is also crazy about cricket...'), Llama-3.1 predicted it sarcastic and gave the reason as *'The phrase "Yeh bhee cricket kee deewani hain" translates to "You are also a cricket fanatic" in English. However, the use of the word **"bhee"** (also) in a sentence suggests that the speaker is being sarcastic, implying that the person is extremely obsessed with cricket, which is often considered a stereotypical trait in Indian culture. The tone of the sentence suggests that the speaker is mocking or ridiculing the person's enthusiasm for cricket.'*

F Experimental configurations:

In this section, we reported the details of experimental setups and model configurations.

F.1 Experimental set-up:

To conduct our experiments, we divided the code-mixed datasets into a training(80%), validation(10%) and test(10%) set, with stratified sampling. We kept the test set the same as in (Khandelwal et al., 2018) and (Swami et al., 2018). Additionally, we constructed several augmented training sets comprising native language task samples. Since no Hindi humour and sarcasm datasets were readily available, we created synthetic datasets by translating some portions of English datasets using Google Translator API. We randomly sampled data points from English and candidate translation samples. Further, we ensured an equal number of samples to be selected from both classes in each augmented training set to avoid complexities related to class imbalance during training. Table 8 reports the label distribution of the considered training sets, validation sets and test sets. For our Multi-Task Learning (MTL) experiment, we trained our model by feeding samples batchwise, with each batch containing samples from multiple tasks. This approach avoided sequential training to prevent bias towards any specific task, especially the one pro-

cessed last. As a result, samples from humor, sarcasm and hate detection appeared in the same batch. For cases where the task label was missing, we used an ignore label ('999'). Figure 4 provides an example dataframe. The main architecture of our MTL framework is presented in Figure 3. This is a BERT-based architecture consisting of 12-layers. The whole model is divided into two halves: i) bottom 8 layers common for all tasks and ii) top 4 layers for task-specific training. Here, the upper module consisting of top 4 layers is thus replicated n times, where n is the number of tasks added with one. This extra upper module is for shared features among the semantically related tasks. Finally, for each task, a gating mechanism combines the shared module output with the task-specific module output, to get the final logits. For clarity in each step, we also provided the related pseudocode in Algorithm 1.

Partition	Dataset	# Humor	# Non-Humor
Train	Code-mixed	1407	953
Augment English	Col	1180	1180
	POTD	1180	1180
	HaHa	1180	1180
	16000	1180	1180
Augment Hindi	Col (translated)	1180	1180
	POTD (translated)	1180	1180
	HaHa (translated)	1180	1180
	16000 (translated)	1180	1180
Val	Code-mixed	176	119
Test	Code-mixed	176	119

Partition	Dataset	# Sarcasm	# Non-Sarcasm
Train	Code-mixed	403	3797
Augment English	NHD	2100	2100
	iSarc	1067	1067
	SC-V2	2100	2100
Augment Hindi	NHD (translated)	2100	2100
	iSarc (translated)	1067	1067
	SC-V2 (translated)	2100	2100
Val	Code-mixed	50	475
Test	Code-mixed	50	475

Partition	Dataset	# Hate	# Non-Hate
Train	Code-mixed	1661	2914
Augment train	HCIn	1416	1416
	HASOC	2261	2261

Table 8: Dataset statistics considered for the native sample mixing experiments with their train-val-test split. Notation: # for number of samples,'translated' for translated Hindi.

Algorithm 1 MultiTaskModel Algorithm

Input: Text input tokens ($input$)
Given: BERT encoder ($BERT$), Task-wise last four layers of BERT module ($module_{task_1}, module_{task_2}$), Gating scheme ($gate$)
Output: Logits ($comb_{task_1}, comb_{task_2}$).

function MULTITASKMODEL($input$)
  $bert_{hidden} \leftarrow BERT(input)$
  $bottom \leftarrow bert_{hidden}[8]$
  for $layer$ in $module_{task_1}$ **do**
    $task_1 \leftarrow layer(bottom)$
  end for
  for $layer$ in $module_{task_2}$ **do**
    $task_2 \leftarrow layer(bottom)$
  end for
  $comb_{task_1} \leftarrow gate(bert_{hidden}, task_1)$
  $comb_{task_2} \leftarrow gate(bert_{hidden}, task_2)$
  return $comb_{task_1}, comb_{task_2}$
end function

To implement the soft-parameter sharing, we introduced a regularization term in the joint loss function. The joint loss function for the MultiTaskModel (when number of tasks, i.e., $T = 2$) is defined as:

$$\mathcal{L}_{\text{joint}} = L_1 + L_2 + \lambda \cdot \|\mathbf{W}_{task_1} - \mathbf{W}_{task_2}\|_2$$

where:

- L_1: Loss for Task 1,
- L_2: Loss for Task 2,
- λ: Regularization strength,
- $\mathbf{W}_{task_1}$: Weight matrix from the last or second last layer of Task 1,
- $\mathbf{W}_{task_2}$: Weight matrix from the last or second last layer of Task 2.

The term $\|\mathbf{W}_{task_1} - \mathbf{W}_{task_2}\|_2$ represents the L_2-norm (Euclidean distance) between the weight matrices of the two tasks' specific layers, enforcing soft parameter sharing between tasks (Rotman and Reichart, 2019).

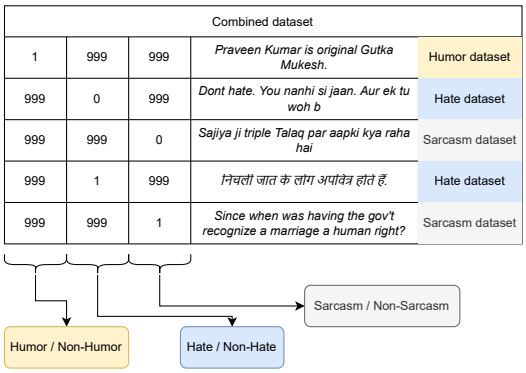


Figure 4: Restructured dataset considered for MTL-based experiments.

F.2 Model configurations:

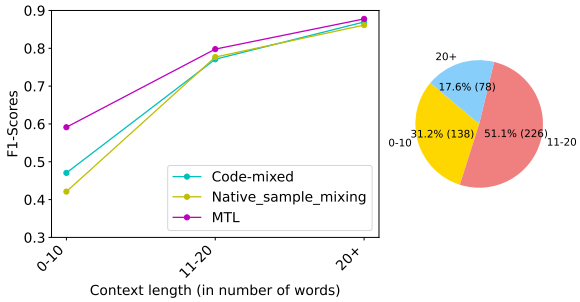
We conducted all our experiments on a single NVIDIA A100 GPU card. We presented our considered set of VMLMs and their respective versions in Table 10. This set includes both open-source and closed-source source VMLMs. For closed source models like GPT-4, it's important to note that their weights might change in the future as they are updated and improved. We conducted all our experiments with these models during the period from September, 2024 to November, 2024. We used a default set of hyperparameters for the VMLMs across runs to maintain consistency in results.

Parameters	Values
Learning rate	{2e-6, 2e-5, 2e-4, 3e-3, 9e-3, 1e-2}
Optimizer	SGD, AdamW
Gamma value (Scheduler)	0.9, 0.8
Loss	Weighted CE
Weights (loss)	$\left[\frac{N}{P+N}, \frac{P}{P+N}\right]$
Batch size	16, 32, 64
Sequence length	64, 128, 248
Patience (Early stop)	4
Regularization strength	{0, 5e-1, 5e-2, 5e-3, 5e-4}
Number of few shots	{0, 2, 4, 8, 12}

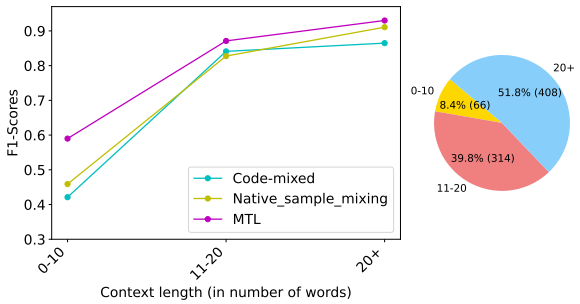
Table 9: Model configurations for experiments. Notation: 'P' for number of positive sample and 'N' for number of negative sample.

VMLM	Version
Gemma	`Telugu-LLM-Labs/Indic-gemma-7b-finetuned-sft-Navarasa-2.0`
Aya Expanse	`CohereForAI/aya-expanse-8b`
Llama-3.1	`meta-llama/Llama-3.1-8B-Instruct`
GPT-4	`ChatGPT`

Table 10: Model versions in VMLMs.



(a) Code-mixed humor detection



(b) Code-mixed sarcasm detection

Figure 5: Performance analysis with increasing context length. Here, the corresponding pie-chart represents the distribution of context length.

G Ablation study & error analysis:

G.1 Ablation study:

In this section, to analyze the role of gating component within the multi-task learning model, we removed it to compare performance and the outcomes are presented in Table 4. Key observations include:

1. **Spelling errors:** For instance, consider the sarcastic statement: *"@flypigmk uski g**d mein dum hai.. agar kisi aur ke g**d mein nahi hai to uske baap ka kya jaat hai... #sarcasm with #g**d"* (**Gloss:** @flypigmk, he has strong a**... if someone else doesn't have the a**, what does that say about his father's caste... #sarcasm with #a**). Here, the MTL model with gating is able to detect the typo error, *'jata'*(**Gloss:** goes) is misspelled as *'jaat'*(**Gloss:** caste), however the MTL without gating got confused.

2. **Shorter context:** For example, in the humorous sample: *"Sir @arvindkejriwal AAP karen to chamatkaar, BJP kare to balatkaar."*(**Gloss:** Sir @arvindkejriwal, If AAP does it, it's a miracle, if BJP does it,

Sl No	Sample	Translated English	CM			NSM			MTL		
			M1	M2	M3	M1	M2	M3	M1	M2	M3
1	.@shashitharoor sir Kejriwal power cut nahi karenge to bill kam kaise hoga? **(Humor)**	.@shashitharoor sir Kejriwal power cut nahi karenge to bill kam kaise hoga?	×	×	×	×	×	×	×	✓	✓
2	Musalmaano ka intolerance kuch zyada hi badh raha hai.. Par Media gaalia sirf Hindu ko deti hai **(Non-humor)**	The intolerance among Muslims seems to be increasing excessively... But the media abuses only Hindus.	✓	✓	✓	×	✓	×	×	×	×
3	Kehte hain Agar kisi cheez ko dil se chaaho to puri kayanat usey tumse milane ki koshish mein lag jaati hai. #dada is back #ipl4 #srk #irony **(Non-sarcasm)**	They say if you truly desire something from the heart, the whole universe conspires to make it happen. #dada is back #ipl4 #srk #irony	✓	✓	✓	×	×	✓	×	×	×
4	Culturally rich honge..par gavaaro ki basti bhi mera desh. #SuSaid #irony #india **(Sarcasm)**	They might be culturally rich, but my country is also a haven for the uneducated. #SuSaid #irony #india	×	×	×	✓	×	×	✓	✓	×
5	@RahulBose1 Fir bhi mera bharat maahan. #Sarcasm **(Non-sarcasm)**	@RahulBose1 Yet, my India is still great. #Sarcasm	✓	✓	✓	×	×	×	×	✓	×

Table 11: Selected examples for various cases reported under error analysis. Here, the '✓', and the '×' denote correct and incorrect classification by the corresponding model, respectively. Notation: CM for code-mixed, NSM for native sample mixing, MTL for Multi-Task Learning; M1 for mBERT, M2 for XLM-R and M3 for MuRIL. The columns under **CM** reported the results when the models were trained with only code-mixed samples and the columns under **MTL** reported the results of the best performing MTL model for each task.

it's a rape.), the gating mechanism overtook the model without gating by detecting humorous contrast using the rhyming words *"chamatkaar"* (**Gloss:** miracle) and *"balatkaar"* (**Gloss:** rape) within shorter context. In a similar way, in the sarcastic statement: *"@iamyasaar Flop graphy Ki baat Goti fan ke muh se?? #Irony"*(**Gloss:** @iamyasaar Talking about flop graphy from the mouth of a Goti fan?? #Irony), gated model was able to detect the ironic situation where a person who is perceived to be a fan of something unsuccessful is commenting on another failure, within such shorter context.

G.2 Error analysis:

To better understand the models' errors, we conducted a qualitative error analysis by examining some correctly and incorrectly classified samples, as presented in Table 11. We observed the following:

- For the ironic humor in *Sl. No. 1*, the humor arises from the switch between the political promise (*'no power cuts'*) and the ironic consequence (*'high bills'*) of the situation. Most of the models failed on it, except XLM-R$_{MTL}$ and MuRIL$_{MTL}$ as it had source of knowledge from other tasks like sarcasm.

- *Sl. No. 2* shows how all the MTL models struggled with non-humorous sample related to religious domain containing keywords like *'intolerance'* and *'gaalia'*, likely due to task interference from the hate detection task, where these keywords are often used in hateful contexts.

- In *Sl. No. 3* and *4*, despite keywords like *#Sarcasm*' and *#irony*', models trained on code-mixed data accurately predicted non-sarcastic contexts, whereas NSM and MTL models failed.

- MTL models effectively captured sarcasm in hateful contexts by combining hate detection with other tasks. For example, in *Sl. No. 5*, the word *'gavaaro'* (**Gloss:** uneducated) conveys explicit hate and MTL models identified the sarcastic tone in it.

H Examination of translated Hindi data:

In this section, we reported our qualitative investigation of code-mixed and translated Hindi samples. This investigation led to two crucial

observations. We first observed that most of the code-mixed humour and sarcasm samples are Hindi dominated. Secondly, for many samples humour and sarcasm got lost when they were translated from English samples. To showcase it, we reported some examples of Hindi translation obtained using Google Translate API in Figure 6. The humor and sarcasm in the English samples often rely on wordplay, puns, ironic and idiomatic expressions that may not have direct equivalents in Hindi. The translated versions attempt a literal translation, losing the subtleties, play on words and cultural context present in the original English samples. In the first humor example, the Hindi translation fails to capture the wordplay of *"denial"* and *"Nahhhh"*(onomatopoeic word used for sheep), resulting in a literal and less humorous translation. In the second humor example, the Hindi translation fails to capture the play on words related to the news about Samsung phones "blowing up", as the literal translation does not convey the intended humor. In a similar way for sarcasm examples, the Hindi translation lacks the subtlety and incongruity necessary for sarcasm, as it straightforwardly conveys the situation without emphasizing the ironic tone. This leads to a drop in degree of sarcasm of the translated Hindi version. Thus, the Hindi translations of English (especially more for humor) data samples did not preserve the native cultural context. This analysis emphasizes the need for a more precise context aware translation method. Since, translated Hindi samples didn't preserve the humorous and sarcastic context, we decided to use only English samples for further experiments.

Humor translations:

1. What does a sheep in denial say? Nahhhh
   Hindi Translation: इनकार में भेड़ क्या कहती है? नहह

2. Ever since the news came out about Samsung.... Their phones have been blowing up.
   Hindi Translation: जब से सैमसंग के बारे में खबर सामने आई है.... उनके फोन धड़ाम हो रहे हैं।

Sarcasm translations:

1. "You're never going to have a white boyfriend, are you Jess? Nah, don't think so."
   Hindi Translation: "आप कभी भी एक श्वेत प्रेमी नहीं होने जा रहे हैं आप जेस हैं? नाह ऐसा नहीं सोचते"

2. babe stop i'm about to gleek
   Hindi Translation: बेब स्टॉप मैं ग्लीक के बारे में हूँ

Figure 6: Translated samples of Hindi humor and sarcasm directly from the native English dataset.