

Harmful Meme Detection in Indic Languages: Dataset Curation and Baseline Development

B Keerthan Varma, K Dinesh Siddhartha, K Hemanth,
M Lakshmi Manasa, Praveen Kumar, R Bhavana, V Venkat Akhilesh Naik

Indian Institute of Technology Gandhinagar

{23110068, 23110168, 23110170, 23110193, 23110257, 23110274, 23110348}@iitgn.ac.in

Abstract

Memes are increasingly used for communication on social media, but a growing number contain harmful content, including misinformation, stereotypes, and hate speech. In this work, we curate large-scale Indic meme datasets across Hindi, Telugu, Tamil, and Kannada and develop baseline models for zero-shot harmful meme detection. We integrate IndicBERT for textual feature extraction alongside multimodal embeddings and implement a hybrid annotation pipeline combining LLM-assisted reasoning with manual verification. We evaluate our baselines using Macro-F1 to address dataset imbalances and provide insights into multilingual harmful meme detection.

1 Introduction

1.1 Motivation

Memes combine text and visuals and spread rapidly online. While often humorous, many memes are harmful, spreading misinformation or reinforcing stereotypes. Detecting harmful memes is essential for safer digital spaces.

1.2 Relation with NLP

Harmful meme detection requires understanding textual and visual cues, semantic analysis, contextual interpretation, and reasoning. NLP enables extraction and interpretation of textual signals, which are critical for accurate classification.

1.3 Problem Type

This is a binary classification task: predicting whether a meme is harmful or harmless.

2 Feedback and Novelty

2.1 Feedback

Since Presentation 1, our mentor provided guidance on dataset design, model selection, and evaluation strategies. Key feedback includes:

- **Dataset Expansion:** Increase dataset size to ~8,000 memes per Indic language (Hindi, Telugu, Tamil, Kannada).
- **Annotation Strategy:** Use LLMs for annotation; LLMs struggled with subtle harmful intent, so we tried implementing a hybrid pipeline with manual verification.
- **Textual Feature Extraction:** Replace DistilBERT with IndicBERT for Indian languages. MOMENTA was initially considered, but we refined baselines using IndicBERT and multimodal embeddings.
- **Handling Dataset Imbalances:** Adopt Macro-F1 and category-wise train-test splits to ensure fair representation of harmful and harmless memes.

2.2 Novelty Introduced

- Curated Indic meme datasets (Telugu, Kannada, Tamil, Hindi).
- Integration of IndicBERT with multimodal embeddings.
- Attempt to gather background knowledge of an image using ConceptNet as an extension to the architecture in MOMENTA paper.

3 Baselines

3.1 State of the Art

Zero-shot harmful meme detection uses large multimodal models (LMMs) with reasoning prompts or retrieval. GPT-4o and Gemini-1.5-Flash are top-performing closed-source models; LLaVA-1.6-34B is a notable open-source baseline used in the paper MIND, which used similarity retrieval and LLM as a judge. The above baselines are only for English text memes but there are no current baselines existing for the indic languages whose datasets, we have made.

3.2 Classic / Training-Based Baselines

Older approaches: multimodal two-stream architectures, late fusion ensembles, and task-specific fine-tuning. These perform well on in-domain data but struggle with out-of-distribution memes.

3.3 Baseline Implementations

No existing implementations for Indic languages. MOMENTA serves as a reference for English.

4 Datasets

- **Dataset Link:** <https://iitgnacin-my.sharepoint.com/my?id=%2Fpersonal%2F23110168%5Fiitgn%5Fac%5Fin%2FDocuments%2FMemes%20Dataset&viewid=2a9076ab%2Dacd6%2D4c2f%2Da710%2Dfcd6fe5e872c>

4.1 Curated Dataset’s Composition and Bias Handling

- **Telugu:** 965 harmful, 1970 harmless (1:2)
- **Kannada:** 573 harmful, 1064 harmless (1:2)
- **Tamil:** 1282 harmful, 1018 harmless (1.26:1)
- **Hindi:** 3776 harmful, 3070 harmless (1.23:1)

Table 1: Dataset composition across train, dev, and test splits for each language.

Language	Train	Dev	Test	Total
Telugu	264	88	88	440
Kannada	147	49	49	245
Tamil	207	69	69	345
Hindi	616	205	205	1026

Bias mitigation: Macro-F1 and category-wise splits to ensure fair evaluation.

4.2 Bottlenecks Encountered

- LLM-based automatic labeling was inaccurate for subtle harmful content.
- Limited publicly available Indic meme datasets.
- Manual curation and annotation were labor-intensive.
- A considerable amount of the memes weren’t much clear in terms of resolution because of which text wasn’t properly identified, thus, We had to discard them from the training set, which was a waste of time.

4.3 Dataset Extension and Differences

- **Telugu/Kannada:** Handcrafted from scratch, first open-source datasets.
- **Tamil:** Curated from TamilMemes dataset, removed non-harmful troll content.
- **Hindi:** Combined Memotion3 and MIMIC, removed English-only memes, manual annotation, and added web-scraped memes.

5 Experimental Setting

5.1 Overview of Experimental Setting

The goal of the experiment was to classify each meme as **harmful** or **non-harmful**. We curated datasets across four Indic languages and split them into three subsets with the ratio of 60% training, 20% development, and 20% testing. We are using on 15% of the dataset from each language as the total dataset for the training, testing and validation of the model because of computational constraints. To minimize potential bias, especially in memes depicting specific individuals, memes referring to the same person were distributed proportionally across the train, dev, and test splits. This ensured fair representation and avoided data leakage through identity overlap.

The experimental pipeline integrates multiple modalities:

- A **CLIP image encoder** for extracting global image-level features.
- An **IndicBERT encoder** for processing and representing the meme text.

- A **VGG-19 model** to extract local region-based features using proposal regions.
- A **ConceptNet-based model** to capture relevant background knowledge and context.

These embeddings were fused using self-attention and cross-attention mechanisms (Code used from MOMENTA paper), followed by a neural network that condensed the multimodal representations into a single binary output corresponding to the harmfulness classification.

5.2 Model Training Setup

We used pre-trained models such as Indic-Bert, CLIP, VGG-19, and ConceptNet as frozen encoders, while the final neural layers were trained on our curated dataset.

The development set was used for hyperparameter tuning, optimizing for both Macro-F1 and accuracy.

Hyperparameters:

The optimizer- Adam

learning rate - $1e-3$ and $1e-5$

epochs- 20

batch size- 32

Training was performed using a single GPU on colab, which required training on a subset of the dataset to maintain computational feasibility.

5.3 Reliability and Resource Constraints

Although the results are partially constrained by limited training data and hardware, the experiment maintains interpretability and reliability through its design. Each module—text, background knowledge, and image feature encoders—produces interpretable intermediate representations that can be traced and analyzed. Despite resource constraints, the multimodal flow and the interaction between these components provide confidence in the model’s decision-making process.

5.4 Models and Approaches Used

Due to limited time, we implemented a single multimodal approach inspired by the **MOMENTA** paper and extended it with ConceptNet embeddings for background knowledge integration. The approach combines textual, visual, and conceptual embeddings using attention-based fusion.

5.5 Approach Variations and Differences

Currently, only one approach has been implemented due to time constraints. Future iterations will involve experimenting with different fusion mechanisms and encoder combinations to compare performance across architectures.

5.6 Architecture and Parameters

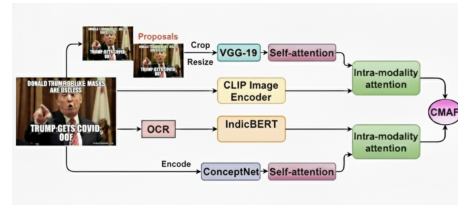


Figure 1: Architecture of the Pipeline

Parameter Count- **3.74 M** parameters

5.7 Hyperparameter Tuning Strategy

Hyperparameter tuning was conducted using the development set. The primary selection criteria were development accuracy and Macro-F1 score. Hyperparameters such as learning rate, batch size, and number of attention heads were adjusted iteratively to achieve optimal validation performance.

5.8 Parameter Search Method

We employed a manual search strategy guided by development performance rather than an exhaustive grid search due to computational limitations. Future work may involve exploring random search or Bayesian optimization for efficient hyperparameter selection.

5.9 Ablation Study Table Template

Table 2: Ablation study on validation performance across val datasets.

Dataset	LR	Batch	Val Macro-F1	Val Acc.
Hindi Dataset	1e-3	32	0.41	0.45
Hindi Dataset	1e-5	32	0.37	0.41
Tamil Dataset	1e-3	32	0.32	0.36
Tamil Dataset	1e-5	32	0.39	0.43
Telugu Dataset	1e-3	32	0.35	0.39
Telugu Dataset	1e-5	32	0.40	0.44
Kannada Dataset	1e-3	32	0.35	0.39
Kannada Dataset	1e-5	32	0.37	0.43

Table 3: Ablation study on test performance across test datasets.

Dataset	LR	Batch	Test Macro-F1	Test Acc.
Hindi Dataset	1e-3	32	0.39	0.43
Hindi Dataset	1e-5	32	0.36	0.40
Tamil Dataset	1e-3	32	0.30	0.33
Tamil Dataset	1e-5	32	0.38	0.41
Telugu Dataset	1e-3	32	0.34	0.38
Telugu Dataset	1e-5	32	0.39	0.42
Kannada Dataset	1e-3	32	0.33	0.37
Kannada Dataset	1e-5	32	0.37	0.41

5.10 Evaluation Metrics

We used both **Macro-F1** and **Accuracy** as the primary evaluation metrics. Macro-F1 was chosen to handle class imbalance between harmful and harmless memes, while accuracy measures overall correctness. These two metrics together provide a balanced view of the model’s classification performance.

5.11 Additional Metrics

No additional metrics were explored, as the combination of Macro-F1 and Accuracy sufficiently captured both class-wise balance and total prediction performance.

For the viva demonstration, we will showcase the following pipeline: an input meme will pass through the image, text, and background-knowledge encoders, and the model will output the predicted harmfulness label.

6 Project Management

6.1 Novel Solution Development

We aimed to build upon the existing **MOMENTA** architecture, as it represents a strong multimodal benchmark and provides clear interpretability of the reasoning process behind harmful meme detection. After mentor discussions, we incrementally refined the idea and attempted to integrate Concept-Net an extension that incorporates background knowledge into the model. Instead of using the original MOMENTA pipeline’s precomputed background vectors, our approach integrates a knowledge retrieval stage that extracts and embeds background concepts related to entities or objects detected in the meme image. These ConceptNet-derived embeddings were then combined with textual and visual features within the multimodal fusion layer, enhancing contextual understanding and enabling the model to reason about implicit harm or stereotypes.

6.2 Computational Resources

The project required moderate computational support for both training and inference. Experiments were executed on a single workstation with the following configuration:

- **GPU:** 7x 15 GB, T4 GPU (colab)
- **CPU:** 7 x 1 cores (colab)
- **RAM:** 7 x 12.7 GB RAM (colab)
- **Storage:** 20 GB (used out of 700 GB)

To optimize for limited compute resources, model training was performed on a subset of the dataset, with frozen encoders for heavy pre-trained networks (CLIP, VGG-19, ConceptNet), and only the attention and classification layers were trained.

6.3 Task Distribution

- **Keerthan Varma:** Dataset curation, Documentation, report writing, and Experimentation pipeline.
- **Dinesh Siddhartha:** Dataset curation, ConceptNet integration, and Experimentation pipeline.
- **Hemanth:** Dataset curation, Data preprocessing, and ConceptNet integration.
- **Lakshmi Manasa:** Dataset curation, Evaluation metrics computation, ablation analysis,

and report writing.

- **Praveen Kumar:** Dataset curation, Model training, Image Encoder Integration and Conceptnet Integration.
- **Bhavana:** Dataset curation, Documentation, report writing, result summarization and Experimentation pipeline.
- **Venkat Akhilesh:** Dataset curation, Hyperparameter tuning, Model training and IndicBert Integration.

7 Implementation

What we implemented

- Full pipeline: dataset-prep scripts, OCR/text encoding, train-test-dev 60/20/20 splits, image encoders, training loop (existing).
- Encoders used: CLIP (global image), VGG-19 (local proposals), IndicBERT (text), ConceptNet embeddings (background knowledge), MOMENTA (fusion and attention modules).

Novelty delivered

- Primary novelty: one of the first curated multilingual Indic meme datasets (Hindi/Telugu/Tamil/Kannada).
- Model novelty: integration of ConceptNet embeddings into a MOMENTA paper’s pipeline.

Code and reproducibility

- Repository containing the codebase: <https://github.com/Akhilesh348/Multilingual-Memes-Classification-Harmful-Non-Harmful-.git>.

Evaluation

- Metrics used: Macro-F1 (to handle class imbalance) and Accuracy.

8 Results and Findings

The model shows consistent performance across languages, capturing multimodal cues despite differences in dataset size and linguistic diversity. However, performance remains below English

models (60% accuracy), indicating room for improvement.

Lower learning rates yield more stable training and better results. Since only about **15% of the full dataset** was used due to computational limits, performance is constrained by the smaller training size.

Validation and test scores are closely aligned, showing decent generalization that can further improve with full dataset training.

Overall, while results are promising, they are still far from optimal — random prediction gives approximately 50% accuracy. Stronger pretraining, richer cross-lingual features, and larger data are needed for better outcomes.

References

- **Momenta:** <https://arxiv.org/pdf/2109.05184>
- **MIND:** <https://arxiv.org/pdf/2507.06908>