Emotion Recognition

IN SARCASTIC CODE-MIXED (HINGLISH) CONVERSATIONS

Team 22

Anjelica Ishita Rathore Suhaani Sachdeva

WHY SARCASM AND EMOTION?

"Without **sarcasm,** life would be as dry as your conversations with a **chatbot—**monotone, predictable, and utterly devoid of personality. **Who wants that?"**

If only AI had an **Eye-roll detector!**

Sarcasm, **EMOTION..**

Text emotion identification and sarcasm detection have grown into important areas of natural language processing (NLP) because of its relevance to many real-world applications like sentiment analysis, social media monitoring, and human-computer interaction. Understanding the hidden meaning of a text that goes against its literal interpretation is what sarcasm detection is all about, while emotion detection seeks to identify the underlying mood or emotional state conveyed in a piece of writing. The complex contextual subtleties of emotion and sarcasm are notoriously difficult for traditional machine learning approaches to grasp, which in turn limits their effectiveness.

Shelke, N. P. P. (2024). Enhanced sarcasm and emotion detection through unified model of transformer and FCNEts. Journal of Electrical Systems, 20(3), 551–565. https://doi.org/10.52783/jes.2982

Humans **Barely** get Sarcasm, so **machines** hardly stand a chance!

Text-Based AI is **clueless** about **Sarcasm**

Imagine telling a chatbot **“wow that is just fantastic”** after it gives you the wrong output and it replies **“Thanks, I try my best”**

Analyzing sentiment or the **emotion** behind **sarcastic expressions** has ***not*** been extensively explored yet it is an important task!
Sarcasm without emotion? that is like a joke without a punchline

largely remained unexplored. A sarcastic expression may have a variety of underlying emotions. For example, “I love being ignored” belies sadness, while “my mobile is fabulous with a battery backup of only 15 minutes!” expresses frustration

Ray, A., Mishra, S., Nunna, A., & Bhattacharyya, P. (2022). A multimodal corpus for emotion recognition in sarcasm. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2206.02119

PROBLEM STATEMENT

Problem Statement

WHAT?

How can we recognize the underlying
- **majority emotion** in
  - ***sarcastic***,
    - ***code-mixed*** *(Hinglish)* conversations using
      - **multimodal** data?

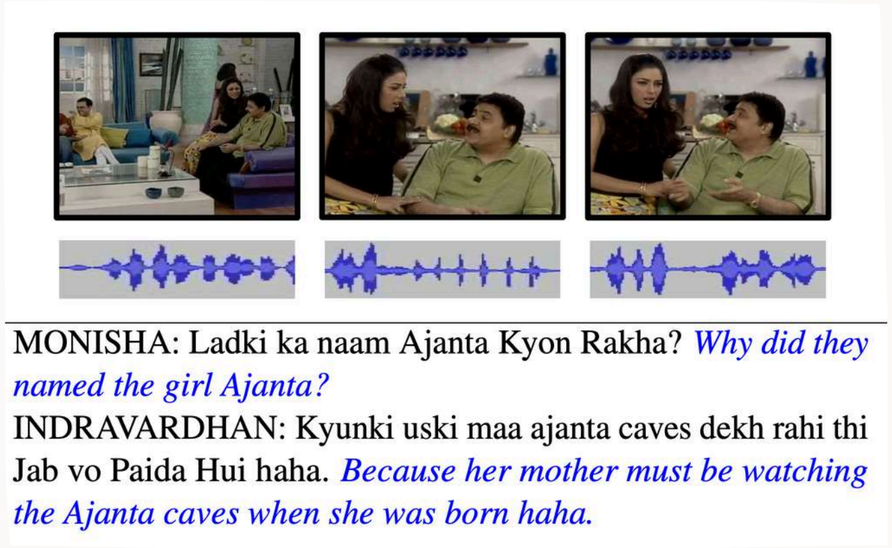
WHY?

Sarcasm in human dialogue flips the *surface meaning*;

- existing systems fail to capture **true emotional intent**, especially when speakers use mixed languages .

IMPACT?

- First system to integrate sarcasm, emotion, and code-mixing in a **multimodal pipeline**.
- Broadens NLP systems to real-world multilingual contexts, reducing bias towards monolingual English.



MONISHA: Ladki ka naam Ajanta Kyon Rakha? *Why did they named the girl Ajanta?*
INDRAVARDHAN: Kyunki uski maa ajanta caves dekh rahi thi Jab vo Paida Hui haha. *Because her mother must be watching the Ajanta caves when she was born haha.*

APPLICATION

- More empathetic chatbots and virtual assistants supporting **Indian multilingual users.**
- Improved content moderation and sentiment analytics for social media in South Asia.
- Enhanced user experience in educational and therapeutic conversational agents.

KUMAR, S., KULKARNI, A., AKHTAR, M. S., & CHAKRABORTY, T. (2022). WHEN DID YOU BECOME SO SMART, OH WISE ONE?! SARCASM EXPLANATION IN MULTI-MODAL MULTI-PARTY DIALOGUES. PROCEEDINGS OF THE 60TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (VOLUME 1: LONG PAPERS). HTTPS://DOI.ORG/10.18653/V1/2022.ACL-LONG.411

LITERATURE REVIEW

Literature Review

Sarcasm *multi-modality* *Code-mixed*

IEEE TRANSACTIONS ON AFFECTIVE COMPUTING. 1

Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations

Manjot Bedi*, Shivani Kumar*, Md Shad Akhtar, and Tanmoy Chakraborty
Dept. of CSE, IIIT-Delhi, India
{manjotb, shivaniku, shad.akhtar, tanmoy}@iiitd.ac.in

***Abstract*—Sarcasm detection and humor classification are inherently subtle problems, primarily due to their dependence on the contextual and non-verbal information. Furthermore, existing studies in these two topics are usually constrained in non-English languages such as Hindi, due to the unavailability of qualitative annotated datasets. In this work, we make two major contributions considering the above limitations: (1) we develop a Hindi-English code-mixed dataset, MaSaC[1], for the multi-modal sarcasm detection and humor classification in conversational dialog, which to our knowledge is the first dataset of its kind; (2) we propose MSH-COMICS[2], a novel attention-rich neural architecture for the utterance classification. We learn efficient utterance representation utilizing a hierarchical attention mechanism that attends to a small portion of the input sentence at a time. Further,**

The proposed model yields the best F1-scores of 71.1% and 82.0% for the sarcasm and humor classification.

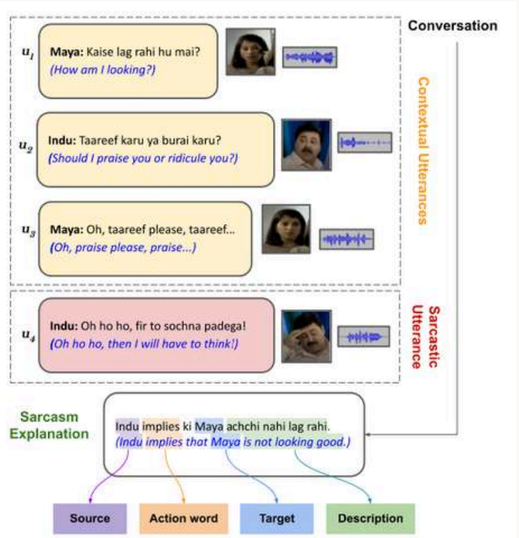
Hierarchical attention mechanism : processes sentences in small, meaningful chunks rather than analysing everything at once.

KBedi, M., Kumar, S., Akhtar, M. S., & Chakraborty, T. (2021). Multi-Modal sarcasm detection and humor classification in Code-Mixed conversations. IEEE Transactions on Affective Computing, 14(2), 1363–1375. https://doi.org/10.1109/taffc.2021.3083522

When did you become so smart, oh wise one?! Sarcasm Explanation in Multi-modal Multi-party Dialogues

Shivani Kumar*, Atharva Kulkarni*, Md Shad Akhtar, Tanmoy Chakraborty
Indraprastha Institute of Information Technology Delhi, India
{shivaniku, atharvak, shad.akhtar, tanmoy}@iiitd.ac.in

ing and generate appropriate responses, simply detecting sarcasm is not enough; it is vital to explain its underlying sarcastic connotation to capture its true essence. In this work, we study the discourse structure of sarcastic conversations and propose a novel task – ***Sarcasm Explanation in Dialogue* (SED)**. Set in a multimodal and code-mixed setting, the task aims to generate natural language explanations of satirical conversations. To this end, we curate **WITS**, a new dataset to support our task. We propose **MAF** (Modality Aware Fusion), a multimodal context-aware attention and global information fusion module to capture multi-modality and use it to benchmark WITS. The



Multimodal Context-Aware Attention : fuse audio-visual cues with text.

Figure 1: Sarcasm Explanation in Dialogues (SED). Given a sarcastic dialogue, the aim is to generate a natural language explanation for the sarcasm in it. *Blue*

Kumar, S., Kulkarni, A., Akhtar, M. S., & Chakraborty, T. (2022). When did you become so smart, oh wise one?! Sarcasm Explanation in Multi-modal Multi-party Dialogues. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). https://doi.org/10.18653/v1/2022.acl-long.411

Literature Review

Emotion ***multi-modality*** *Code-mixed*

A Hybrid Multimodal Emotion Recognition Framework for UX Evaluation Using Generalized Mixture Functions

by Muhammad Asif Razzaq [1,2,†], Jamil Hussain [3,†], Jaehun Bang [4], Cam-Hao Hua [2], Fahad Ahmed Satti [2,5], Ubaid Ur Rehman [2,5], Hafiz Syed Muhammad Bilal [5], Seong Tae Kim [2,*] and Sungyoung Lee [2,*]

Abstract

Multimodal emotion recognition has gained much traction in the field of affective computing, human–computer interaction (HCI), artificial intelligence (AI), and user experience (UX). There is growing demand to automate analysis of user emotion towards HCI, AI, and UX evaluation applications for providing affective services. Emotions are increasingly being used, obtained through the videos, audio, text or physiological signals. This has led to process emotions from multiple modalities, usually combined through ensemble-based systems with static weights. Due to

unimodal emotion recognition and introducing multimodal feature fusion level, and decision level fusion using GM functions. In an experimental study, we evaluated the ability of our proposed framework to model a set of four different emotional states (*Happiness*, *Neutral*, *Sadness*, and *Anger*) and found that most of them can be modeled well with significantly high accuracy using GM functions. The experiment shows that the proposed framework can model emotional states with an average accuracy of 98.19% and indicates significant gain in terms of performance in

The experiment demonstrates that the suggested framework has an average accuracy of 98.19% in simulating emotional states.

The GM function dynamically adjusts the weight of each feature set depending on how relevant or strong it is.

Razzaq, M. A., Hussain, J., Bang, J., Hua, C., Satti, F. A., Rehman, U. U., Bilal, H. S. M., Kim, S. T., & Lee, S. (2023b). A hybrid multimodal emotion recognition framework for UX evaluation using generalized mixture functions. Sensors, 23(9), 4373. https://doi.org/10.3390/s23094373

A Multitask Framework for Sentiment, Emotion and Sarcasm aware Cyberbullying Detection from Multi-modal Code-Mixed Memes

Krishanu Maity*
Indian Institute of Technology Patna
Patna, India
krishanu_2021cs19@iitp.ac.in

Prince Jha*
Indian Institute of Technology Patna
Patna, India
princekumar_1901cs42@iitp.ac.in

sarcastic, and multi-modality (image + text). The current work is the first attempt, to the best of our knowledge, in investigating the role of sentiment, emotion and sarcasm in identifying cyberbullying from multi-modal memes in a code-mixed language setting. As a contribution, we have created a benchmark multi-modal meme dataset called *MultiBully* annotated with bully, sentiment, emotion and sarcasm labels collected from open-source Twitter and Reddit platforms. Moreover, the severity of the cyberbullying posts is also investigated by adding a *harmfulness* score to each of the memes. The created dataset consists of two modalities, text and image. Most of the texts in our dataset are in code-mixed form, which captures the seamless transitions between languages for multilingual users. Two different multimodal multitask frameworks

Maity, K., Jha, P., Saha, S., & Bhattacharyya, P. (2022). A Multitask Framework for Sentiment, Emotion and Sarcasm aware Cyberbullying Detection from Multi-modal Code-Mixed Memes. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. https://doi.org/10.1145/3477495.3531925

Literature Review

Emotion in Sarcasm

multi-modality

Text-Audio-Video

A Multimodal Corpus for Emotion Recognition in Sarcasm

Anupama Ray, Shubham Mishra, Apoorva Nunna, Pushpak Bhattacharyya
IBM Research India, Department of Computer Science and Engineering, IIT Bombay, India
anupamar@in.ibm.com, shubham101mishra@gmail.com, {apoorvanunna, pb}@cse.iitb.ac.in

and *Illocutionary* (Camp, 2012). **Propositional sarcasm** needs context information to be able to detect whether it's sarcasm or not. For example: "your plan sounds fantastic!" may seem non-sarcastic if the context information is not present (Zvolenszky, 2012). **Embedded sarcasm** has an embedded incongruity within the utterance; thus, the text itself is sufficient to detect sarcasm. For example: "It's so much fun working at 2 am at night". **Like-prefixed sarcasm** as the name suggests uses a like-phrase to show the incongruity of the argu-

al., 2017). **Illocutionary sarcasm** is a type of sarcasm that bears the sarcasm in the non-textual cues, and the text is often the opposite of the attitude captured in the audio or video modality. (Zvolenszky, 2012) give an example of rolling eyes while saying "Yeah right" being a

Abstract

While sentiment and emotion analysis have been studied extensively, the relationship between sarcasm and emotion ha largely remained unexplored. A sarcastic expression may have a variety of underlying emotions. For example, "I love bein

Detecting the emotion behind a sarcastic expression is non-trivial yet an important task. We undertake the task of detecting the *emotion in a sarcastic statement*, which to the best of our knowledge, is hitherto unexplored. We start with the recently released multimodal sarcasm detection dataset (MUStARD) pre-annotated with 9 emotions. We identify and correct 343

he perceived emotion of the speaker. While *valenc*
ndicates the extent to which the emotion is positive o
negative, *arousal* measures the intensity of the emotio

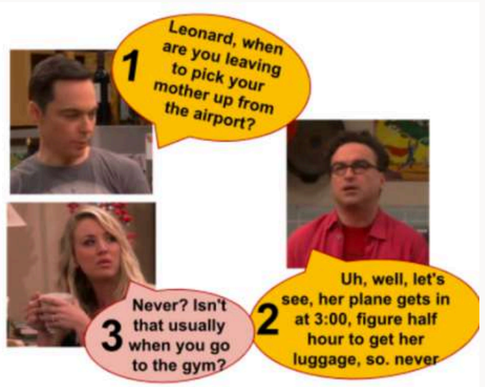


Figure 1: Example to show that different explicit anc
implicit emotions in sarcasm. *Explicit* = *Surprise*, anc
Implicit = *Ridicule*; Sarcasm Type: Embedded; Valenc

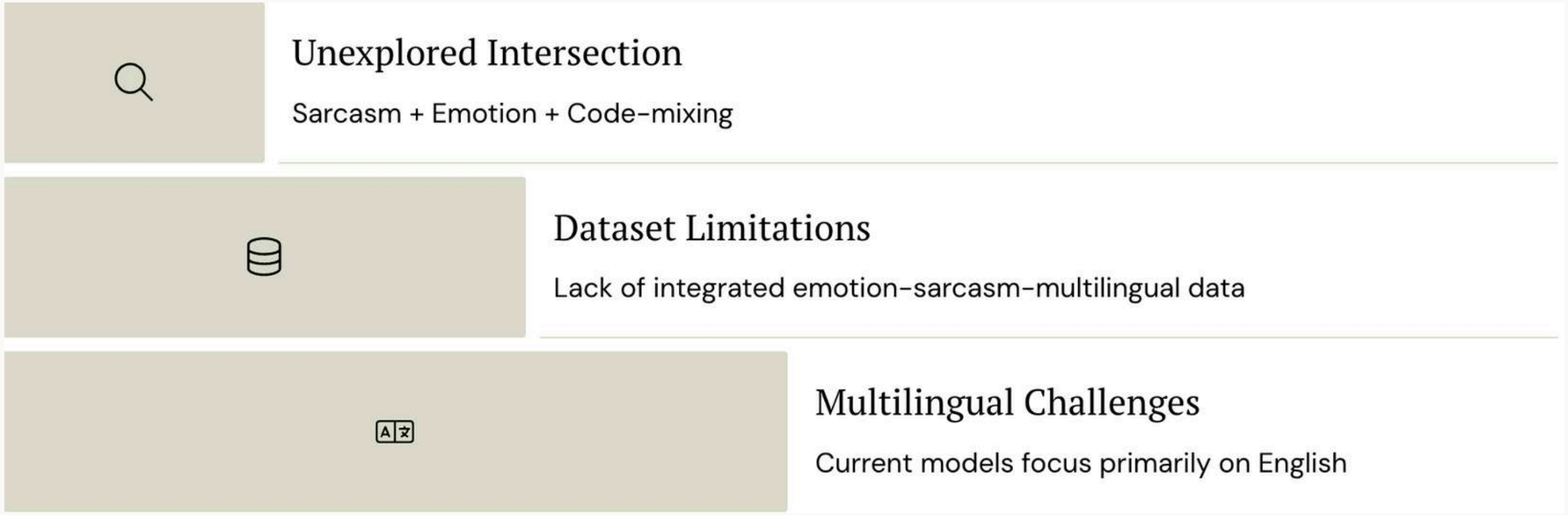
Ray, A., Mishra, S., Nunna, A., & Bhattacharyya, P. (2022, June 1). A multimodal corpus for emotion recognition in sarcasm. ACL Anthology. https://aclanthology.org/2022.lrec-1.756/

Literature Review

Study	Techniques	Relevance	Score/Metrics
A Multimodal Corpus for Emotion Recognition in Sarcasm	• **Modalities** Used : **Text**, **Audio** & **Video**. • **Embeddings** : BART-large for text , audio features (MFCC) for audio & ResNet-152 for video. • **Collaborative gating architecture** : fusion of text, audio, and video features.	• Addresses emotion recognition in sarcasm using multimodal data, for the English language only.	**Weighted F1 score** for emotion in sarcasm: • **~40.8%** (Text+Audio+Video without context) • **~41.5%** (Text+Audio+Video with context)
When did you become so smart, oh wise one?! Sarcasm Explanation in Multi-modal Multi-party Dialogues	• **WITS** dataset : with natural language explanations for sarcastic conversations. • Uses Modality Aware Fusion (MAF) module, which includes **Multimodal Context-Aware Attention** (MCA2) and Global Information Fusion (GIF)	The paper focuses on explaining sarcasm rather than explicit emotion recognition metrics.	• **MAF-TAVB model**: Best performer - multimodal fusion yields significant gains vs. text-only baselines. • **Source vs. Target**: Source identification accuracy +13% (91.07%) with multimodality
Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations	• Proposed neural architecture called MSH-COMICS utilizes LSTMs. • **Hierarchical attention mechanism** • **Features** : Textual, Acoustic. • **Gating mechanism** : Multimodal fusion.	• **MaSaC** dataset. • Relevant for its work on code-mixed (Hinglish) data and sarcasm detection.	• **MSH-COMICS superiority**: Text-only **F1 scores** of 69.8% (sarcasm) and 79.3% (humor), outperforming DialogRNN by ~3% and ~4.5% respectively • **Multimodal** : Text+Acoustic fusion achieved **F1 scores** of **68.6%** (sarcasm) and 81.4% (humor) in joint training
A Multitask Framework for Sentiment, Emotion and Sarcasm aware Cyberbullying Detection from Multi-modal Code-Mixed Memes	• **Meme Dataset** : MultiBully, annotated with bully, sentiment, emotion, and sarcasm labels. • **Frameworks** : BERT+ResNET-Feedback & CLIP-CentralNet. • Simple concatenation to **merge textual and visual features.**	• Code-mixed Hindi-English text and images. • Provides performance metrics for sarcasm and emotion recognition using multi-modal frameworks.	• **Sarcasm Detection :** an **Accuracy** of 74.17 & **F1 score** of 74.11 • **Emotion Recognition** : an Accuracy of 71.85 & F1 score of 71.77

RESEARCH GAP

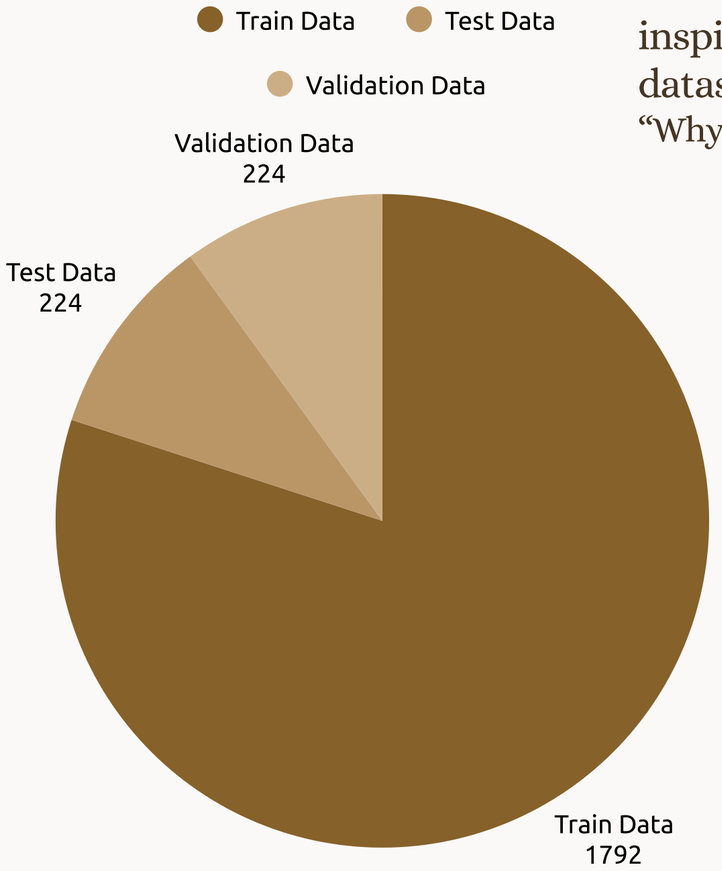
what we found...



This problem has not been explored before. In the best of our knowledge, our idea would be the first in this field.
Hence, emotion labeling has been done manually by us, for each text data point.

OUR DATASET

Dataset



inspired by WITS dataset 2022
"Why is this Sarcastic"

Text + Videos

Citation
Kumar, S., Kulkarni, A., Akhtar, M. S., & Chakraborty, T. (2022). When did you become so smart, oh wise one?! Sarcasm explanation in multi-modal multi-party dialogues. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 5956–5968). Association for Computational Linguistics. https://aclanthology.org/2022.acl-long.411

Why WITS?

The WITS dataset is a multi-modal, code-mixed, sarcasm detection dataset in dialogues with code-mixed explanations.

Primary reason of using WITS → Code-Mixed Data(Hinglish)
This dataset is built on top of another code-mixed dataset, called MaSac.

Text Transcripts

Video Clips

Our final Dataset

MaSac Dataset

Our multi-modal sarcasm and humor classification dataset is based on the video clips of the popular Indian comedy TV show *'Sarabhai vs. Sarabhai'*. The show resolves around the day-to-day life of five family members, namely, Indravardan (aka Indu), Maya, Saahil, Monisha, and Roshesh, with a few infrequent characters. Each scene of the show involves conversation among two or more speakers, and based on the speaker, we split the conversation into utterances. In all, we extract more than 15K utterances from 400 scenes spread across 50 episodes. We refer to the conversation (or sequence of utterances) in each scene as a standalone dialog. For each utterance in the dialog, we assign appropriate sarcasm (*sarcastic* or *non-sarcastic*) and humor (*humorous* or *non-humorous*) labels. Thus, the context for any utterance is restricted to the conversation in the current dialog only. We employed three annotators for assigning sarcasm and humour labels to each utterance. Finally, we aggregate the annotations using majority voting. We also calculate the Cohen Kappa inter-rater agreement score for the annotations. The average score for humor classification is 0.654, whereas for sarcasm detection it is 0.681.

ments. Out of 14,000 utterances in the train set, the number of sarcastic and humorous utterances are 2,748 and 5,054, respectively. Similarly, the test set comprises 391 sarcastic and 740 humorous utterances. Table II also lists the word distribution for the Hindi-English code-mixed input. MaSaC consists of ~36,000 Hindi and ~3,000 English words.

Bedi, M., Kumar, S., Akhtar, M. S., & Chakraborty, T. (2023). Multi-modal sarcasm detection and humor classification in code-mixed conversations. IEEE Transactions on Affective Computing, 14(2), 1363–1375. https://doi.org/10.1109/TAFFC.2021.3083522

WITS Dataset

the proposed task, we curate a new dataset named WITS, where we augment the already existing MASAC dataset (Bedi et al., 2021) with explanations for our task. MASAC is a multimodal, multi-party, Hindi-English code-mixed dialogue dataset compiled from the popular Indian TV show, 'Sarabhai v/s Sarabhai'[2]. We manually analyze the data and clean it for our task. While the original dataset contained 45 episodes of the TV series, we add 10 more episodes along with their transcription and audio-visual boundaries. Subsequently, we select the sarcastic utterances from this augmented dataset and manually define the utterances to be included in the dialogue context for each of them. Finally, we are left with 2240 sarcastic dialogues with the number of contextual utterances ranging from 2 to 27. Each of these instances is manually

KUMAR, S., KULKARNI, A., AKHTAR, M. S., & CHAKRABORTY, T. (2022). WHEN DID YOU BECOME SO SMART, OH WISE ONE?! SARCASM EXPLANATION IN MULTI-MODAL MULTI-PARTY DIALOGUES. PROCEEDINGS OF THE 60TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (VOLUME 1: LONG PAPERS). HTTPS://DOI.ORG/10.18653/V1/2022.ACL-LONG.411

WITS++

This is our dataset, **built on WITS**, consisting of train, test and validation *(text and video)* instances.

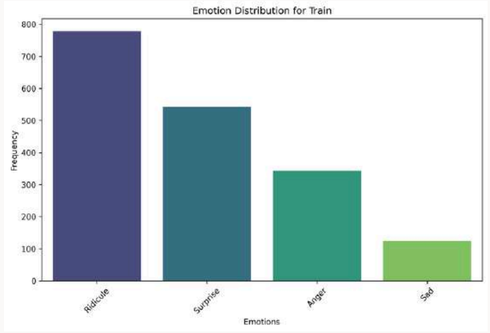
We have **manually annotated** these text instances (with emotions), and categorized them into 4 main emotion types: Anger, Surprise, Sad and Ridicule.

NOTE: the following data points had to be removed due to discrepancies in video and text transcripts:Train-1184, 1201, 1385 , 1591; Test- 87

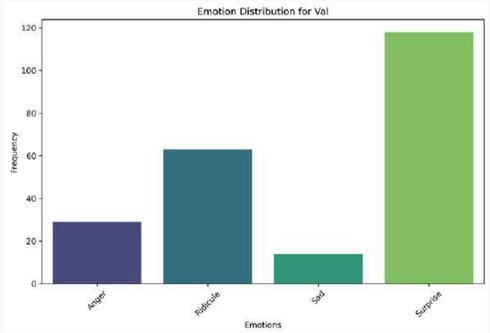
DISTRIBUTIONS
of emotions

Emotion Distributions

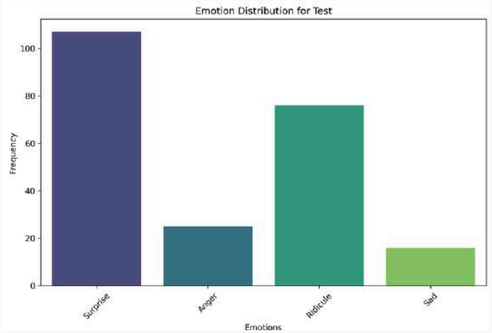
train



val



test



Emotions Used-
Anger,
Surprise,
Ridicule,
Sad.

→ Each EMOTION label signifies the majority emotion, which is (most often) leading to Sarcasm.

→ We assessed each instance individually, and then assigned a final label to the instance based on majority voting.

All discrepancies were cross-checked with Plaksha Counsellor, Dr. Shalini

Annotation Process

The WITS dataset *does not* contain **Emotion Labelings**, that are required for our project.

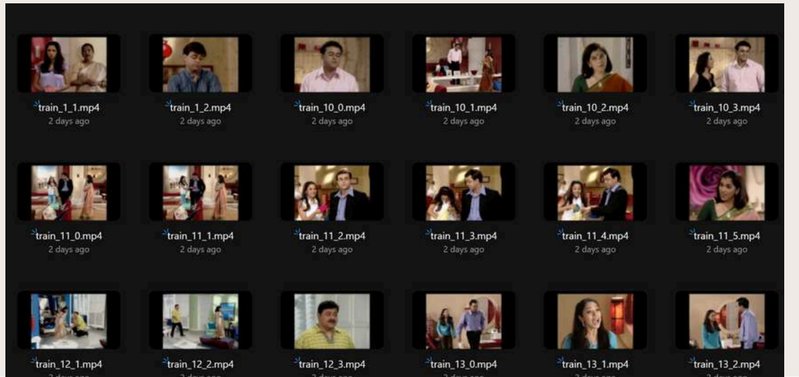
Therefore, using the text and video modalities (data), we annotated *all the text point*s (train, test and validation data) **ourselves**.

Following is what the final JSON file looked like...

Text


```
"1582": {
  "episode_name": "17-Maya's bet with Monisha",
  "target_speaker": "MAYA",
  "target_utterance": "Kaise sawaal kar rahe ho, Monisha ki neend haraam hogi to mujhe maza aaega? Of course mujhe maza aaega!",
  "context_speakers": ["MAYA", "RADHABAAI"],
  "context_utterances": [
    "Iss baar mai Monisha se 10000 rupay vasool karke rahungi. Neend haraam ho jaegi bechaari ki, paise dete hue!",
    "Aap ko maza aaega?"
  ],
  "english_explanation": "Maya jokes that she will be happy extorting money from Monisha.",
  "code_mixed_explanation": "Maya majaak karti hai ki usse Monisha se paise vasool karne mein bohot maja aayega.",
  "sarcasm_target": "MONISHA",
  "sarcasm_type": "I",
  "start_time": "00:06:43",
  "end_time": "00:06:56",
  "emotion": "Ridicule"
},
"1583": {
```


Videos



FEATURE PRE-PROCESSING

Feature Pre-processing

TEXT

Taking cue from our *literature review*, we used **FastText multilingual word embedding model** to create our text embeddings.
This model is considered good for *morphologically rich languages* like **Hindi** (or **Hinglish**).

FastText supervised model , trained on our *code-mixed (**train set**) data* generated **100-dimensional vector embeddings** for each instance of our text (target + context utterances combined into one). This was done for Train, Test and Validation datasets.

C. Feature Extraction

We employ pre-trained FastText multilingual word embedding model [21, 22] and Librosa [47] tool for the textual and acoustic representations, respectively. For each token in

2) Uni-modal evaluation – Textual: Similar to the acoustic modality, we also perform experiments with only the textual modality. In total, we perform four variants, i.e., *LSTM*(T_{avg}), *LSTM*(*H-ATN*^{U}), *LSTM*(T_{BERT}) and *LSTM*(*H-ATN*^{U}) + *C-ATN*^{D}. The first variant is a vanilla LSTM based classification model trained on the textual utterance embeddings - which is computed as the mean of FastText multilingual embeddings of constituent words, and is represented as T_{avg}. The second variant, *LSTM*(*H-ATN*^{U}), is similar to the first except that the textual utterance embeddings is computed utilizing the utterance-level hierarchical attention module. The third variant

Bedi, M., Kumar, S., Akhtar, M. S., & Chakraborty, T. (2023). Multi-modal Sarcasm Detection and Humor Classification in Code-mixed Conversations. IEEE Transactions on Affective Computing, 14(2), 1363-1375. https://arxiv.org/pdf/2105.09984

Feature Pre-processing

VIDEOS

Owing to our Literature review, we created keyframes of the video clips and then passed them through **ResNet-50** and **a custom CNN** in order to get the embeddings per keyframe and then average the embeddings for each instance.

Step1 : Getting the Keyframes

- Literature used **Katna :** It gave us less control over extraction, & did not produce enough keyframes for processing.
- Switched to **OpenCV** using SSIM.

Step2 : Passing the keyframes to a CNN:

- Passing the keyframes through both CNNs.
- Average the vectors.

Video Modality

In order to extract visual features from the videos, we have used a pool5 layer of pre-trained ResNet-152 (He et al., 2016) image classification model. To improve the video representation and reduce noise, we extracted the key frames to be passed to ResNet-152, instead of feeding in information from all of the frames. Key frame extraction is widely used in the vision community and is defined as the frames that form the most appropriate summary of a given video (Jadon and Jasim, 2019). We used an open source tool called Katna[4], to perform key-frame extraction. For final feature vectors we average the vectors of each key frame of an instance (context and utterance) extracted from ResNet-152. The size of final video feature representation in $d_v = 2048$.

Ray, A., Mishra, S., Nunna, A., & Bhattacharyya, P. (2022, June 1). A multimodal corpus for emotion recognition in sarcasm. ACL Anthology. https://aclanthology.org/2022.lrec-1.756/

Feature Pre-processing

THE QUESTION OF BUILDING OUR OWN CNN?

- ResNet 50 : Too many parameters (~25M) → slower inference
- Designed for classification, not lightweight feature extraction
- Risk of overfitting on limited data
- Custom CNN offers full control and interpretability

OUR CNN ARCHITECTURE?

- 4 Convolutional layers with ReLU and MaxPooling
- Final layer : AdaptiveAvgPool2d → Fully Connected
- Outputs : 2048-D feature vector per keyframe

ML METHODOLOGY

ML Methodology

LITERATURE REVIEW INSPIRED

- Following the steps that we took from the literature review we created a model-**S**hallow **N**eural **N**etwork relying on **A**ttention based **F**usion **M**echanism.

- This takes in text embeddings that are first standardized and reduced using PCA.

- These reduced features are separately encoded through dedicated neural encoders—a text encoder and a video encoder—comprising linear layers, batch normalization, and dropout for regularization.

- The encoded modalities are then combined using an attention-based fusion mechanism that dynamically weighs the contribution of each modality. The fused representation is passed through a final classifier to predict one of the target emotions (sad, angry, surprise, ridicule).

SNN-AFM

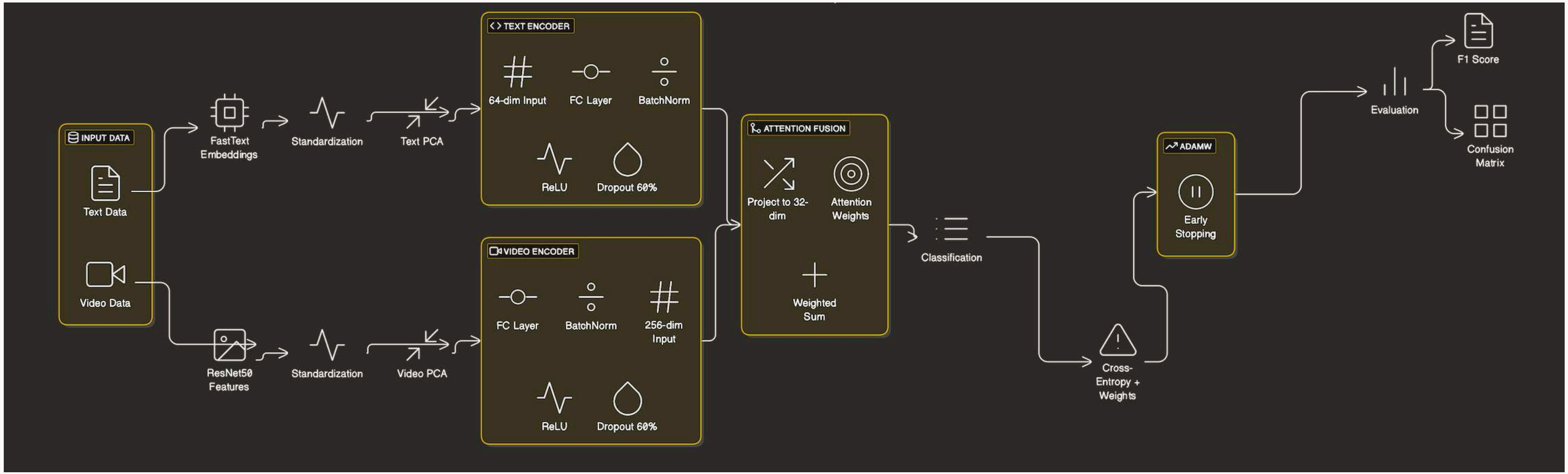
```
class AttentionFusion(nn.Module):
    def __init__(self):
        super().__init__()
        self.text_projection = nn.Linear(TEXT_HID_DIM, FUSION_OUT_DIM) #transforms the 32-dimensional t
        self.video_projection = nn.Linear(VIDEO_HID_DIM, FUSION_OUT_DIM) #Transforms the 64-dimensional
        self.attention = nn.Linear(FUSION_OUT_DIM * 2, 2)# Produces the attention weights that determin


    def forward(self, text_features, video_features):
        # Projecting features
        text_proj = torch.relu(self.text_projection(text_features))
        video_proj = torch.relu(self.video_projection(video_features))
        # Concatenating for attention calculation
        combined = torch.cat([text_proj, video_proj], dim=1)
        attention_weights = torch.softmax(self.attention(combined), dim=1)
        # Applying attention weights to the respective features
        fused = attention_weights[:, 0].unsqueeze(1) * text_proj + \
                attention_weights[:, 1].unsqueeze(1) * video_proj
        return fused

class SimplifiedModel(nn.Module):
    def __init__(self, text_dim, video_dim, num_classes):
        super().__init__()
        self.text_encoder = SimplifiedTextEncoder(text_dim)
        self.video_encoder = SimplifiedVideoEncoder(video_dim)
        self.fusion = AttentionFusion()
        self.classifier = nn.Linear(FUSION_OUT_DIM, num_classes) # Input: The 32D fused representation;
        self.dropout = nn.Dropout(DROPOUT_RATE)

    def forward(self, text_input, video_input):
        text_features = self.text_encoder(text_input)
        video_features = self.video_encoder(video_input)
        fused_features = self.fusion(text_features, video_features)
        fused_features = self.dropout(fused_features)
```

MODEL ARCHITECTURE



ML Methodology

RANDOM FOREST CLASSIFIER

We created a simple code, with **Random Forest Classifier**, to see its performance on our dataset, and to evaluate if a complex architecture was needed.
This approach has proved that a **simple architecture is enough** for our problem.

After loading all the necessary files, we performed **simple (axis 1 → column-wise) concatenation** of the text and video features *for each instance*.
Following this, the model was trained on **Train Data** having *concatenated embeddings* and evaluated on validation and test sets.

BEST WORKING MODEL

Test Accuracy - 52.9 %
Weighted F1 score - 0.53 (Better than existing models, operated on similar problem)

Loading all embedding files → Concatenation of embeddings (text + video) → Evaluate using Random Forest Classifier

```
Evaluating on test data...
Test Accuracy: 0.5291

Test Classification Report:
              precision    recall  f1-score   support

           0       0.32      0.42      0.36        24
           1       0.64      0.46      0.54       107
           2       0.49      0.71      0.58        76
           3       1.00      0.31      0.48        16

    accuracy                           0.53       223
   macro avg       0.61      0.47      0.49       223
weighted avg       0.58      0.53      0.53       223
```

PERFORMACE METRICS

WHY F1-SCORE

The formula for **F1-score** is:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

F1-score can be interpreted as a ***weighted average*** or ***harmonic mean*** of **precision** and **recall**, where the relative contribution of precision and recall to the **F1-score** are equal. **F1-score** reaches its best value at 1 and worst score at 0.

What we are trying to achieve with the **F1-score** metric is to find an ***equal balance between*** **precision** and **recall**, which is extremely useful in most scenarios when we are working with **imbalanced datasets** *(i.e., a dataset with a non-uniform distribution of class labels)*.

OUR INSIGHTS

- Comparable performance despite **excluding audio** modality.
- Simpler models yield **better results** compared to complex models.
- Ridicule and Surprise emotion detected with highest accuracy among four classes, whereas Sad was quite underrepresented.
- Visual cues carry significant emotional signal in code-mixed sarcasm.

SNN-AFM vs Random Forest Classifier

SNN-AFM

```
--- Model Evaluation ---
Test Accuracy: 0.5112

Classification Report:
              precision    recall  f1-score   support

           0       0.35      0.46      0.40        24
           1       0.63      0.44      0.52       107
           2       0.54      0.66      0.59        76
           3       0.25      0.38      0.30        16

    accuracy                           0.51       223
   macro avg       0.44      0.48      0.45       223
weighted avg       0.54      0.51      0.51       223
```

ACCURACY → 51.12%
WEIGHTED F1-SCORE → 0.51

RANDOM FOREST CLASSIFIER

```
Evaluating on test data...
Test Accuracy: 0.5291

Test Classification Report:
              precision    recall  f1-score   support

           0       0.32      0.42      0.36        24
           1       0.64      0.46      0.54       107
           2       0.49      0.71      0.58        76
           3       1.00      0.31      0.48        16

    accuracy                           0.53       223
   macro avg       0.61      0.47      0.49       223
weighted avg       0.58      0.53      0.53       223
```

ACCURACY → 52.91%
WEIGHTED F1-SCORE → 0.53

OUR NUMBERS

SNN-AFM

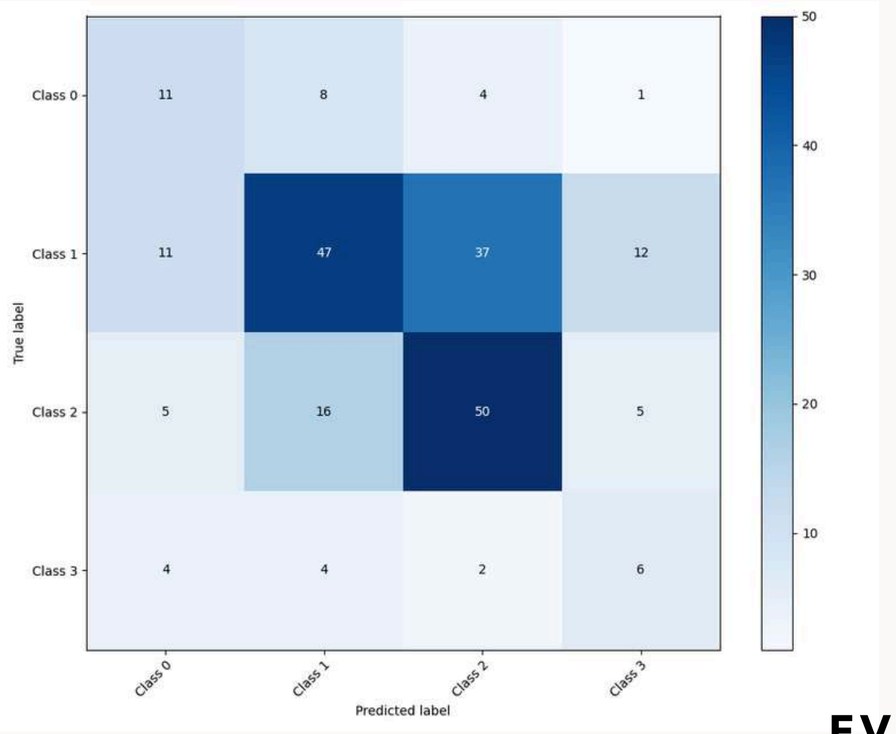
F1 score *weighted avg* 0.51

0.59 *Ridicule*

0.52 *Surprise*

0.40 *Anger*

0.30 *Sad*



RANDOM FOREST CLASSIFIER

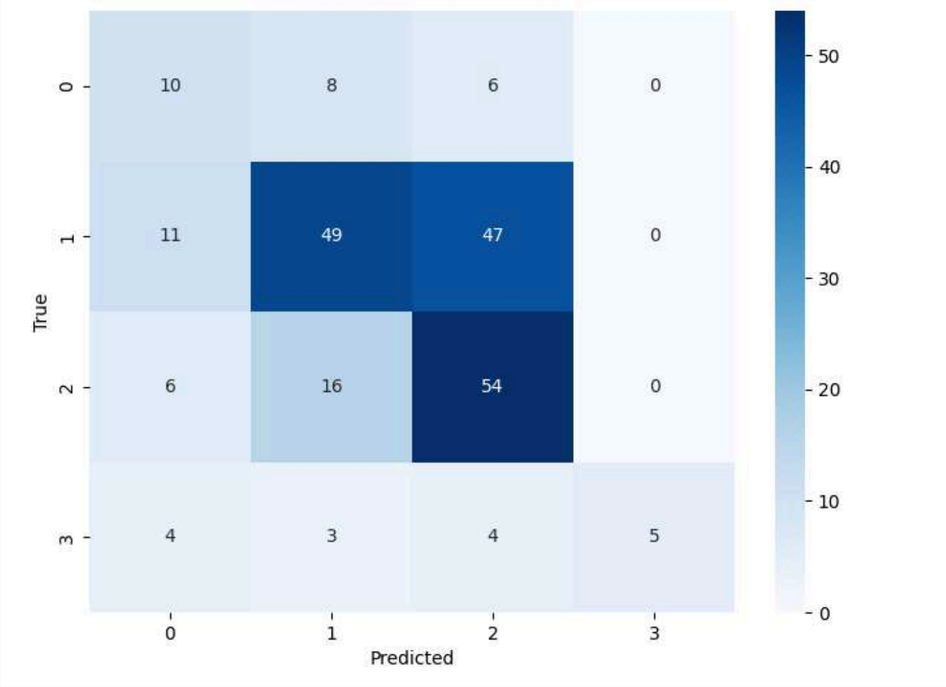
F1 score *weighted avg* 0.53

0.58 *Ridicule*

0.54 *Surprise*

0.36 *Anger*

0.48 *Sad*



Anger (0)
Surprise (1)
Ridicule (2)
Sad (3)

EVALUATED ON TEST DATA

Resnet-50 vs our CNN

RESNET-50

```
--- Model Evaluation ---
Test Accuracy: 0.5112

Classification Report:
              precision    recall  f1-score   support

           0       0.35      0.46      0.40        24
           1       0.63      0.44      0.52       107
           2       0.54      0.66      0.59        76
           3       0.25      0.38      0.30        16

    accuracy                           0.51       223
   macro avg       0.44      0.48      0.45       223
weighted avg       0.54      0.51      0.51       223


Confusion Matrix:
[[11  8  4  1]
 [11 47 37 12]
 [ 5 16 50  5]
 [ 4  4  2  6]]
```

BASIC CNN

```
--- Model Evaluation ---
Test Accuracy: 0.5067

Classification Report:
              precision    recall  f1-score   support

           0       0.35      0.46      0.40        24
           1       0.62      0.43      0.51       107
           2       0.54      0.66      0.59        76
           3       0.24      0.38      0.29        16

    accuracy                           0.51       223
   macro avg       0.44      0.48      0.45       223
weighted avg       0.54      0.51      0.51       223


Confusion Matrix:
[[11  8  4  1]
 [11 46 37 13]
 [ 5 16 50  5]
 [ 4  4  2  6]]
```

DEPLOYABILITY

Potential deployment scenario at Plaksha:

- Vaani : This model could potentially be employed in Vaani, Plaksha's chatbot, to better assist multi-lingual (Hinglish, specifically) audience.

- Customer Service chatbots : In order to better provide assistance, our model could be used in this field, to understand the customers better, and give them a good experience.

Scalability Challenges

- Expanding to additional language pairs
- Adapting to different cultural sarcasm styles
- Integration with existing systems

Future Improvements

- Including audio modality for enhanced performance
- Expanding the dataset to more diverse languages
- Addressing underrepresented emotion classes, expanding the model to more classes.

THANK YOU