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

Cyberbully Detection Project

Devika Singh, Divyanshu Bathla, Gaurav B, Nirban Das

Indian Institute of Guwahati

Contents

Introduction

Cyberbullying and Memes I

- **Cyberbullying:** Involves serious, intentional, and repetitive acts of cruelty towards others through digital platforms such as Instagram and Twitter.
- **Memes as a Medium:** While individual text or images might not fully convey information, their combination in memes provides a richer context and can be used for cyberbullying.
- **Consequences:** The effects of cyberbullying can lead to significant issues, including anxiety, depression, and emotional distress.
- **Importance of Early Detection:** Identifying cyberbullying early is crucial for implementing effective measures to address and mitigate its impact.
- **Advancing Detection Methods:** Given that memes consist of multi-modal content (image + text), utilizing advanced multi-modal models and NLP techniques can yield promising results for more accurate detection and analysis.

Cyberbullying and Memes II

- Various multimodal framerowks have been proposed
- BERT + ResNet Feedback and CLIP-CentralNet
- Cyberbully Detection (CD), Sentiment Analysis(SA), Emotion Recognition(ER), Sarcasm Detection(SAR)
- A new dataset has been made called MultiBully, each data point having a harmfullness score

Related Work

Works on Monolingual Datasets I

Study	Data Source	Methodology/Tools	Results
Dinakar et al. [1]	4,500 YouTube comments	Binary classifiers (SVM, Naive Bayes)	SVM: 66.70%, Naive Bayes: 63%
Reynolds et al. [2]	Formspring.me	Weka toolkit, C4.5 decision tree	78.5% accuracy
Djuric et al. [3]	Yahoo Finance comments	Paragraph2vec, CBOW	80.01% accuracy
Balakrishnan et al. [4]	Twitter users	Psychological characteristics, Machine Learning	91.7% accuracy
Paul et al. [5]	Formspring (12k posts), Twitter (16k posts), Wikipedia (100k posts)	BERT-based framework (cyberBERT)	State-of-the-art results; BERT pooled output (CLS token) dimension: 768

Works on mixed code dataset I

Study	Data Source	Methodology/Tools	Results
Kumar et al. [6]	18k tweets, 21k Facebook comments (Hindi-English code-mixed)	Aggression-annotated corpus	Data used for aggression annotation
Bohra et al. [7]	4,575 tweets (code-mixed)	SVM classifier with features: word n-grams, punctuations, character n-grams, hate lexicon, negation words	71.7% accuracy
Satyajit et al. [8]	Hindi-English code-mixed corpus	Deep learning approach with domain-specific word embedding	12% improvement in F1 score over base model
Maity et al. [9]	Code-mixed Indian language dataset	Deep learning architectures: BERT, CNN, GRU, capsule networks	79.28% accuracy

Works on Sentiment, Emotion and Sarcasm aware Multitasking I

Study	**Focus**	**Methodology/Tools**	**Results**
Saha.T et al. [10]	Multi-modal tweet act classification (TAC)	Multi-task ensemble adversarial learning framework	TAC significantly outperforms uni-modal and single-task TAC variants
Soumitra et al. [11]	Emotion classification on suicide notes	Multi-task learning architecture with external knowledge	Improved overall performance
Dushyant Singh et al. [12]	Sarcasm detection analysis	Multi-task framework with Inter-segment and Intra-segment attention mechanisms	Analyzed effects of sentiment and emotion on sarcasm detection
Lewis et al. and Maity et al. [13,14]	Cyberbullying detection from Hinglish code-mixed text	Attention-based multitask models	Investigated sentiment and emotion for cyberbullying identification

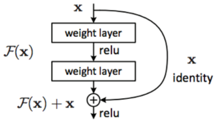
Works on Meme Datasets I

Study	Data Source	Methodology/Tools	Results
Kiela et al. [15]	Benchmark multimodal meme dataset	Visual-BERT	69.47% testing accuracy
Gomez et al. [16]	Tweets with image and text (MMHS150K)	Manual annotation for hate speech	Multimodal dataset for hate speech detection
Bharathi et al. [17]	Multi-modal (Image+Text) Meme Dataset (MultiOFF)	Early fusion approach for image and text modalities	Compared performance with text-only and image-only baselines
Shraman.P et al. [18]	3,544 memes (HarMeme)	Detection of harmful memes (very harmful, partially harmful, harmless)	Dataset for detecting harmful memes and their targets

Priliminaries

ResNet I

- ResNet (Residual Network): A deep neural network architecture introduced in 2015 by He et al..Motivation: Deeper networks should theoretically perform better, but they often suffer from vanishing/exploding gradient problems, making it difficult to train.
- Key Innovation: Introduction of Residual Connections (or skip connections) to allow gradients to flow directly through the network, preventing degradation as the network depth increases.



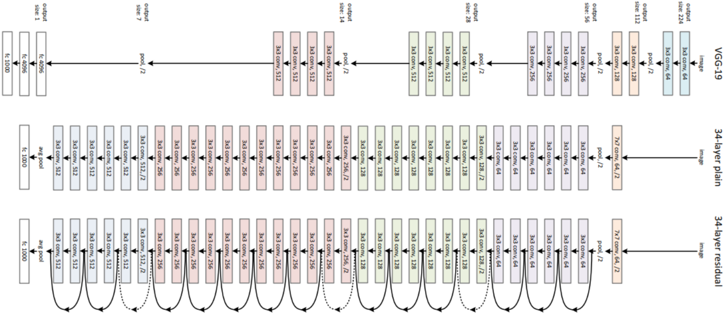
Residual Block

- Standard deep network layer: Input x passes through several convolutional layers.

ResNet II

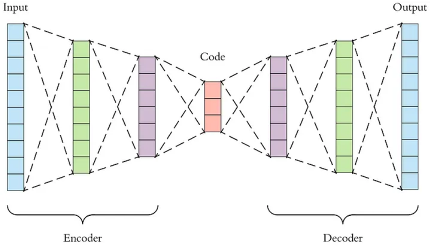
- Residual connection: Input $x$bypasses these layers and is added directly to the output.
- The final output of the block is $F(x) + x$, where $F(x)$ is the transformation applied by the convolutional layers.
  Block Types:
- Identity Block: Shortcut connection skips over layers when input and output dimensions are the same.
- Convolutional Block: Shortcut connection with a convolution is used to adjust dimensions when they differ.

ResNet III



Encoder-Decoder I

- **Encoder-Decoder Framework**: A neural network architecture designed for **sequence-to-sequence tasks**, where input and output sequences can have different lengths.
- **Key Concept**: The **encoder** processes the input sequence into a fixed-size vector (latent representation), while the **decoder** generates the output sequence based on this vector.
- **Applications**: Used in tasks like **machine translation**, **text summarization**, and **image captioning**.



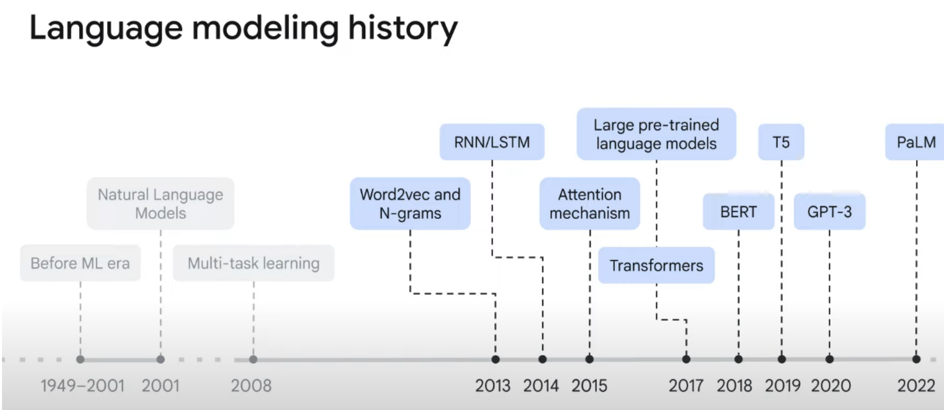
Encoder-Decoder II

- **Encoder Role**:
  - Processes the input sequence and extracts key features.
  - Produces a **context vector** (latent representation).
  - Implemented using RNNs, LSTMs, GRUs, or **transformers**.
- **Decoder Role**:
  - Takes the context vector and generates the output sequence.
  - Operates step-by-step, predicting the next output token.
- **Sequence Length Flexibility**: Can handle variable-length input and output sequences, making it versatile for tasks like translation.
- **Attention Mechanism**: Improves performance by allowing the decoder to focus on specific parts of the input sequence during generation.
- **Variants**:
  - **RNN-based**: Traditional models use RNNs, LSTMs, or GRUs for encoding and decoding.
  - **Transformer-based**: Models like **BERT** and **GPT** use self-attention to improve performance and parallelization.

Encoder-Decoder III

- **Bidirectional Encoders**: Some encoders (e.g., **BiLSTMs**) process sequences in both directions to capture more context.
- **Advantages**:
    - Effective for sequence prediction tasks.
    - Supports flexible design and rich representation learning.

Language Model History

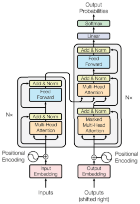


Transformers I

- **Transformers**: Introduced by Vaswani et al. in the paper "Attention is All You Need" (2017).
- **Motivation**: Eliminates the need for recurrence (RNNs) or convolution (CNNs) by relying entirely on the self-attention mechanism.
- **Key Insight**: Leverages **self-attention** to capture dependencies between input tokens, allowing parallel processing and reducing computational complexity.
- **Main Components**:
  - **Encoder-Decoder architecture**.
  - **Self-Attention Mechanism**.
  - **Positional Encoding**.
- Revolutionized natural language processing (NLP) and computer vision tasks, forming the basis for models like **BERT**, **GPT**, and **T5**.
- **Self-Attention**: Allows the model to weigh the relevance of different tokens in the input sequence relative to each other.

Transformers II

- Each token creates three vectors: **Query (Q)**, **Key (K)**, and **Value (V)**.



- The attention score is calculated by taking the dot product of the query and key vectors:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

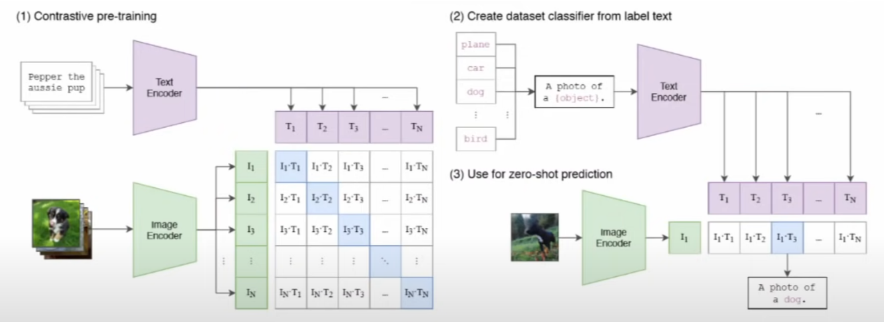
- **Multi-Head Attention**: Combines multiple self-attention layers in parallel, allowing the model to focus on different positions.

Transformers III

- **Scalability**: Self-attention is computationally efficient, making transformers highly parallelizable.
- **Encoder-Decoder Structure**:
  - The **encoder** maps the input sequence into a latent representation.
  - The **decoder** generates the output sequence based on this latent representation.
- **Positional Encoding**: Since transformers lack recurrence, positional encodings are added to input embeddings to represent the order of tokens.
- **Feedforward Networks**: After the multi-head attention layer, fully connected feedforward networks process the output of the attention mechanism.
- **Layer Normalization and Residual Connections**: Each sub-layer in the encoder and decoder uses residual connections and layer normalization for stable training.
- **Applications**: Used in state-of-the-art models for NLP (BERT, GPT, T5) and computer vision (Vision Transformers, ViT).

CLIP I

- CLIP (Contrastive Language–Image Pretraining)
- designed to learn visual representations using natural language supervision. Instead of training models on a predefined set of labeled categories
- learns to associate images with text through contrastive learning.



CLIP II

- The model takes an image and a text as inputs and projects them into a shared multi-modal embedding space where related image-text pairs have high cosine similarity.
- Once trained, CLIP can perform zero-shot classification by predicting which textual description best matches a given image, without being trained directly on the classification task.

Image Encoder:

- CLIP can use different types of image encoders, such as ResNet or Vision Transformer (ViT)
- The image is fed into the encoder, which extracts visual features and outputs an image embedding.
- In the Vision Transformer version, images are divided into patches, and a sequence of patch embeddings is passed through a Transformer to obtain the final representation.

Text Encoder

CLIP III

- The text input is processed by a Transformer model that converts the text into an embedding.
- The text is tokenized and then passed through the Transformer layers, with the output being the representation of the entire text.

BERT I

- BERT (Bidirectional Encoder Representations from Transformers): A transformer-based model introduced by Google AI in 2018.
- Key Idea: Unlike previous models that process text sequentially (left-to-right or right-to-left), BERT is bidirectional, meaning it looks at both directions at the same time.
- **Pretraining and Fine-tuning:** BERT is pretrained on large amounts of text data using unsupervised learning and can be fine-tuned for specific downstream tasks (e.g., text classification, question answering).
- Impact: Achieved state-of-the-art performance on several NLP benchmarks like SQuAD, GLUE, and MNLI.
- Forms the basis of many models like **RoBERTa**, **DistilBERT**, and ALBERT.
- In this paper they had used **mBERT**, which has been trained on many languages, including Hindi and English

Architecture of BERT

BERT II

- **Transformer Encoder Architecture**: BERT uses the **encoder** part of the original transformer architecture.
- **Bidirectional Attention**: Each token attends to all other tokens in the input, capturing rich contextual information.
- **Input Representation**:
  - Combines token embeddings, positional encodings, and segment embeddings.
  - Special tokens: [CLS] for classification tasks, [SEP] to separate sentences.
- **BERT Base**: 12 layers, 768 hidden units, 12 attention heads, 110M parameters.
- **BERT Large**: 24 layers, 1024 hidden units, 16 attention heads, 340M parameters.
- **Pretraining Tasks**:
  - **Masked Language Model (MLM)**: Randomly masks 15
  - **Next Sentence Prediction (NSP)**: Predicts whether two sentences appear sequentially in the original text, aiding tasks like question answering and natural language inference.

BERT III

- Fine-tuning: After pretraining, BERT can be fine-tuned on specific NLP tasks by adding a task-specific output layer.
- **Applications**:
  - Text classification, sentiment analysis, and named entity recognition (NER).
  - Question answering (e.g., SQuAD).
  - Text summarization and machine translation.
- **Bidirectional Nature**: Helps BERT capture deeper context compared to unidirectional models, making it highly effective for language understanding.

Methodology

Frameworks I

- **Multitask Multimodal frameworks:**
  - To identify cyberbullying from memes deep multitask multimodal frameworks have been developed.
  - Feature extraction models: BERT-ResNet Feature Extractor and CLIP Feature Extractor.
  - Multitask frameworks: Feedback Multitask and CentralNet Multitask.
  - Experimented with four model combinations of feature extraction and multitask frameworks:
    - BERT-ResNet + Feedback
    - BERT-ResNet + CentralNet
    - CLIP + Feedback
    - CLIP + CentralNet

BERT-ResNet I

BERT-ResNet Feature Extractor:

- **Text Features :**
    - **Google OCR Vision API5**: Used to extract text from input images.
    - **mBERT**: BERT language model varient chosen for Hindi-English code-mixed memes as it is trained on both languages. It extracts textual features from input text.
    - **Bi-GRU Layer**: Processes mBERT's outputs to capture contextual information. It captures long-term dependencies in word vectors by encoding the input on both forward and backward directions.

$$\overrightarrow{h}_t^i = \overrightarrow{GRU}(w_t^i,\ h_{t-1}^i)\ ,\ \overleftarrow{h}_t^i = \overleftarrow{GRU}(w_t^i,\ h_{t+1}^i) \qquad (1)$$

$$\left[h_t^i = \overrightarrow{h}_t^i, \overleftarrow{h}_t^i\right]$$

- **Image Features :**
    - **ResNet-50**: Used as the base model for image feature extraction due to its strong performance in image classification tasks.

BERT-ResNet II

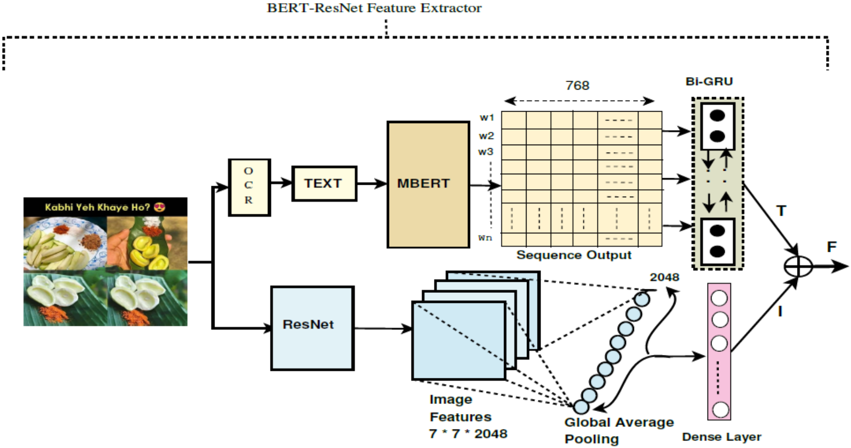
- **Feature Extraction**: Last convoluted features of ResNet-50 ($7 \times 7 \times 2048$) passed through a global average pooling layer resulting in a 2048-dimensional dense vector.
- **Post-Pooling Processing**:
    - Pass previous output through a fully connected (**FC**) layer with 512 neurons, followed by a dropout layer to generate the final image feature vector (I).
- **Feature Concatenation**: text feature vector (BERT+GRU) concatenated with Image feature vector (ResNet+FC) to form the combined feature vector (F).

BERT-ResNet III

Table 5: Model parameters of different feature extractor modules

Features	Model	Type	Output Size
Text	MBERT+BiGRU	MBERT	50×768
		BiGRU	50×512
	CLIP-Text Encoder	BERT	512
Image	ResNet+Dense	ResNet	$7 \times 7 \times 2048$
		GlobalAvgPool	2048
		Dense	512
	CLIP - Image Encoder	Vision Transformer	512

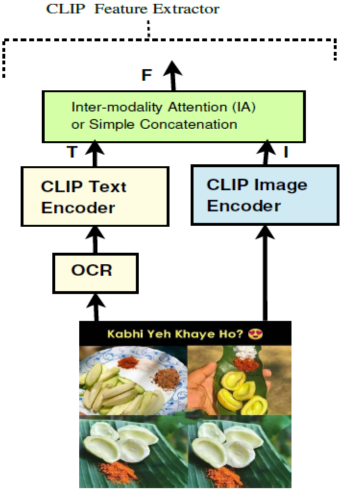
BERT-ResNet IV



CLIP I

- **CLIP (Contrastive Language–Image Pre-training) Feature Extractor:**
  - It is a pre-trained visual-linguistic model, used to encode text–image pairs for semantic understanding of memes.
  - **Pre-trained on**: 400 million image–text pairs from the Internet.
  - **Training Goal**: Predict correct pairings from $N \times N$ possible pairings in a batch of N pairs(image,text). It maximizes cosine similarity for real pairs, minimizes it for incorrect ones.
  - **Optimization**: Symmetric cross-entropy loss based on cosine similarity.
  - **Zero-shot Capabilities**: Due to natural language supervision and wide image range.
  - **Encoders**: Vision Transformer for images and BERT for text.
  - **Extracted Embeddings**:
    - F_I : CLIP image embedding [512 dimensional vector]
    - F_T : CLIP text embedding [512 dimensional vector]
    - I : Meme Image
    - T : OCR-extracted text

CLIP II


CLIP Feature Extractor
F
Inter-modality Attention (IA)
or Simple Concatenation
T
I
CLIP Text
Encoder
CLIP Image
Encoder
OCR
Kabhi Yeh Khaye Ho?

Inter-modal Attention I

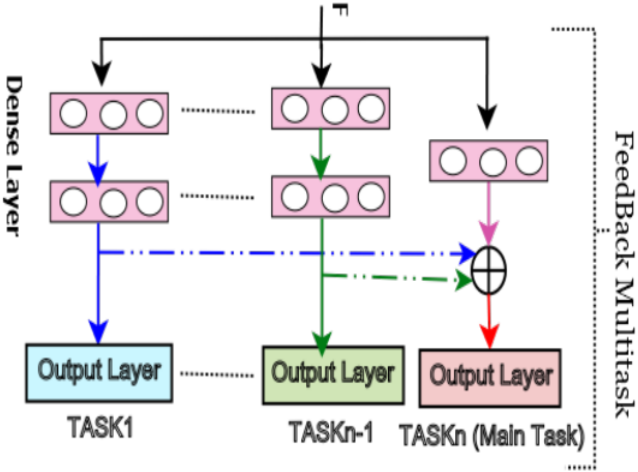
- Text modality is more significant for some memes, while visual modality is more important for others.
- Inter-modal Attention is used to merge textual and visual representations.
- Attention is computed by mapping a query and a set of key-value pairs to an output.
- Outputs of both modalities (T and I) are passed through three fully connected layers:
  - Queries (Q)
  - Keys (K)
  - Values (V)
- The dimensions of Q, K, and V are d_f.
- $\mathbf{IA}_i \in \mathbb{R}^{n_x \times d_f}$.

$$IA_i = softmax(Q_i K_i^T) V_i$$

Feedback Multitask Framework I

- **Framework Overview**:
  - **Multitask Learning**: To learn n tasks simultaneously.
  - **Multimodal Features**: Passed through n task-specific fully connected (FC) layers.
  - **Output Layer**: Each task ends with a softmax layer.
- **Feedback Path**: Feedback from the last FC layers of tasks $T_1, T_2, \ldots, T_n$ to Main Task T_n. It enhances main task performance using features from other tasks.
- **Task-specific Layer Structure**
  - **Secondary Tasks**: Each has two FC layers (100 neurons) + softmax.
  - **Main Task**: Only one FC layer, but concatenate features from other tasks.

Feedback Multitask Framework II



CentralNet Multitask Framework I

- **CentralNet** is a multimodal data fusion network.
- **Reformed as Multitask Framework**: CentralNet is adapted to a multitask setting.
- **Architecture**:
  - n independent task-specific networks.
  - One central network.
  - Task-specific network includes:
    - $n-1$ secondary tasks (ST).
    - One main task (MT).
- **Central Network Function**: Combines features from task-specific networks and its own previous layers.

CentralNet Multitask Framework II

$$MT_{i+1} = \alpha m MT_i + \sum_{k=1}^{n} \alpha s_i^k ST_i^k$$

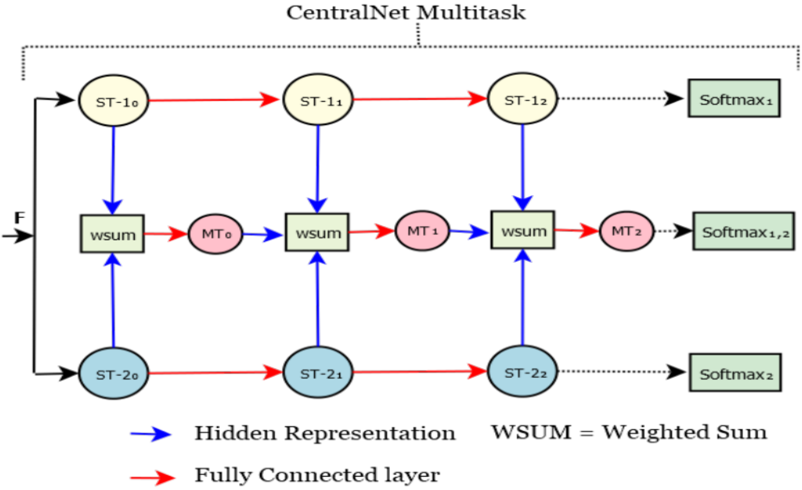
- **Multitask Layer** where:
  - n is the number of task-specific networks
  - αs: Scalar trainable weights.
  - ${ST_i}^{\mathrm{k}}$: Hidden representation of k-th task-specific network at i-th layer.
  - MT_i: Central hidden representation of the main task.
  - Resulting layer MT_{i+1} is fed to an operating layer (dense layer + activation layer).

CentralNet Multitask Framework III

- **Initial Inputs [Central Network]**: Weighted summation of other task-specific initial features.
- **Final Output**: Central network's output is the final prediction for the main task.
- **Model Parameters**: Details are given in Table 6.

Multitask Framework	Task									
	Emotion		Sentiment		Central		Sarcasm		Bully	
	Type	Output Size	Type	Output Size	Type	Output Size	Type	Output Size	Type	Output Size
CentralNet	Dense	512	Dense	512			Dense	512	Dense	512
	Dense	256	Dense	256	Dense	256	Dense	256	Dense	256
	SoftMax	10	SoftMax	3	SoftMax	2	SoftMax	2	SoftMax	2

CentralNet Multitask Framework IV



Loss Function I

- Employ categorical cross-entropy as a loss function to train the network's parameters.

$$L_{CE}(\hat{y}, y) = -\frac{1}{N} \sum_{j=1}^{C} \sum_{i=1}^{N} y_i^j log(\hat{y}_i^j)$$

- $\hat{y}_i^j$ is the predicted label.
- y_i^j is the true label.
- C represents the number of classes.
- N represents the number of memes.

Loss Function II

- The final loss function, Loss, is dependent of N task-specific individual losses as follows

$$Loss = Loss_M + \sum_{k=1}^{n} \beta_k Loss_S^k$$

  - Loss_M is the main-task loss.
  - Loss_S is the secondary task (ST) loss.
  - β ranges from 0 to 1. It defines the loss weights that determine each task's contribution to the total loss.

Results and Discussions

Key Findings I

- **Multi-task Learning Outperforms Single-task:**
  - All multitask models performed better than single-task models.
  - **CLIP+CentralNet** with three auxiliary tasks (Sentiment Analysis, Emotion Recognition, Sarcasm Detection) performed best.
  - **Accuracy Improvement**: +3.18%, **F1 Score Improvement**: +3.1%.
  - Shows that incorporating sentiment, emotion, and sarcasm helps improve **cyberbullying detection (CD)**.

Table 8: Single task results in terms of Accuracy (Acc) and F1 score. FC: Fully connected layer.

Modality	Model	CD		SA		ER		SAR		Harmfulness	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Text (T)	BERT+GRU+FC	61.14	60.73	56.91	54.24	31.23	23.22	59.72	59.12	60.86	60.33
Image (I)	RN+FC	63.36	62.37	58.39	55.61	30.83	23.19	59.39	57.79	62.51	62.14
(T+I) with Concat	BERT+RN+FC	65.04	65.03	60.22	58.82	30.08	26.26	62.20	61.47	65.21	64.66
	CLIP +FC	70.91	70.89	59.8	**59.16**	29.6	**27.96**	63.59	61.24	66.71	65.89
(T+I) with IA	BERT+RN+FC	65.63	65.41	61.02	59.11	30.12	25.39	62.12	62.75	65.28	65.14
	CLIP +FC	70.99	**71.01**	58.96	57.83	26.58	23.32	62.99	**63.80**	66.91	**66.28**

Key Findings II

- **Multimodal (Text + Image) Scenario:**
  - **CLIP + CentralNet** combination was the best performer.
  - Outperformed all other combinations (e.g., BERT-ResNet+Feedback, BERT-ResNet+CentralNet, CLIP+Feedback).
  - Shows that CLIP-CentralNet effectively extracts task-specific features from memes.

Table 7: Experimental results of different multitask variants with unimodal and multimodal settings. CD: Cyberbully Detection, SA: Sentiment Analysis, ER: Emotion Recognition, SAR: Sarcasm, FB: FeedBack, CNT: CentralNet, RN: ResNet, BT-RN: BERT+ResNet, IA: Inter-modal Attention.

Modality	Model	2-Task Variants						3-Task Variants						4-Task	
		CD+SA		CD+ER		CD+SAR		CD+SA+ER		CD+SA+SAR		CD+ER+SAR		CD+SA+ER+SAR	
		Acc	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
Text (T)	BERT+ FB	62.03	61.53	60.40	60.01	61.09	60.82	62.43	62.53	62.96	62.47	61.58	61.38	62.14	62.13
	BERT+CNT	65.94	65.94	65.14	65.29	66.69	65.98	65.34	65.47	65.89	65.92	66.15	66.17	66.61	66.07
Image (I)	RN+FB	64.81	64.51	65.84	65.60	64.13	63.67	64.67	64.31	65.18	64.63	65.18	65.25	64.84	64.92
	RN+CNT	66.89	66.79	66.39	66.45	65.79	67.86	66.42	66.32	66.33	66.27	66.08	66.01	66.43	66.37
T+I with Concat	BT-RN+FB	65.86	65.74	66.52	66.54	65.87	65.82	67.21	67.13	65.23	65.11	65.52	64.96	66.87	66.76
	BT-RN+CNT	69.64	69.36	70.08	69.77	70.20	70.05	69.89	69.64	69.13	68.83	68.46	68.18	69.72	69.44
	CLIP+FB	72.24	72.28	72.16	72.23	72.66	72.68	71.06	71.07	71.85	71.93	71.32	71.35	71.21	71.31
	CLIP+CNT	72.88	72.82	73.03	72.95	72.07	71.96	73.05	72.98	73.05	72.97	73.11	73.02	73.16	73.06
T+I with IA	BT-RN+FB	65.36	65.12	66.82	66.76	66.52	66.41	67.35	67.42	66.15	66.08	65.93	65.12	66.74	66.79
	BT-RN+CNT	73.02	73.05	73.54	73.02	72.22	72.13	73.15	73.07	72.96	72.82	73.28	72.59	73.68	73.53
	CLIP+FB	71.99	72.01	72.75	72.79	71.18	71.18	71.06	71.07	72.33	72.33	71.32	71.35	72.44	72.47
	CLIP+CNT	73.28	73.17	72.66	**72.63**	71.12	71.00	73.79	**73.73**	73.31	73.22	71.85	71.77	74.17	**74.11**

Key Findings III

- **Impact of Combining Modalities:**
  - Simple Concatenation vs. Inter-modal Attention (IA).
  - **Inter-modal Attention** with **CentralNet** consistently outperformed simple concatenation.
  - In contrast, **Feedback** multitask framework didn't show consistent improvement with IA.

- **Effectiveness of Task Combinations:**
  - CD + Sentiment Analysis + Emotion Recognition (CD+SA+ER)
  - Consistently performed better than other combinations like CD+SA+Sarcasm or CD+ER+Sarcasm for multi-modal inputs.
  - Second highest **F1 Score** for CD: 73.73
  - Shows that combining sentiment and emotion provides better context for detecting bullying behavior.

Key Findings IV

- **CentralNet vs Feedback Framework:**
  - **CentralNet** consistently outperforms **Feedback** multitask frameworks.
  - **BERT-ResNet + CentralNet** achieves on average 5% improvement in F1 score over **BERT-ResNet + Feedback** for both multimodal and IA settings.
  - Highlights CentralNet's strength in **multimodal data** fusion and multitask learning.

- **Multimodal vs Uni-modal Performance:**
  - **Multimodal (Text + Image)** variants consistently outperformed uni-modal variants
  - Using both text and image together improved accuracy significantly.
  - In uni-modal settings, image modality performed better than text-only models.

Key Findings V

- **Challenges and Limitations:**
  - Inferior results in **Emotion Recognition** due to the **highly imbalanced** nature of the emotion classes.
  - Despite this, the model's main focus remained on improving **Cyberbullying Detection** (CD).
  - Weighting strategies were used to prioritize the main task.

Conclusion

Overview

- **Task Overview:**
  Introduction of sentiment-emotion-sarcasm aware multimodal cyberbully detection in a codemixed setting.
- **Objective:**
  Explore if sentiment, emotion, and sarcasm labels can enhance cyberbully detection accuracy.

Key Contributions

- **Novel Dataset:**
  - **MultiBully:** A multimodal memes dataset annotated with bully, sentiment, emotion, and sarcasm labels.
  - **Purpose:** To assist in the identification of cyberbullying through nuanced label information.
- **New Architecture:**
  - **CLIP-CentralNet:**
    - An attention-based multi-task multimodal framework.
    - Incorporates ResNet, mBERT, and CLIP.
    - Designed for efficient representation and generalized feature learning across multiple tasks.

Performance Highlights

- **Outperformance:**
  - CLIP-CentralNet outperforms all single-task and uni-modal models.
  - Demonstrates a significant margin in detection accuracy.

References

References I

1. Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. Modeling the detection of textual cyberbullying. In Proceedings of the International Conference on Weblog and Social Media 2011. Citeseer
2. Kelly Reynolds, April Kontostathis, and Lynne Edwards. 2011. Using machine learning to detect cyberbullying. In 2011 10th International Conference on Machine Learning and Applications and Workshops, Vol. 2. IEEE, 241–244.
3. Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In Proceedings of the 24th international conference on world wide web. 29–30.
4. Vimala Balakrishnan, Shahzaib Khan, and Hamid R Arabnia. 2020. Improving cyberbullying detection using Twitter users' psychological features and machine learning. Computers & Security 90 (2020), 101710.

References II

5. Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In International Conference Recent Advances in Natural Language Processing (RANLP). 672–680.
6. Ritesh Kumar, Aishwarya N Reganti, Akshit Bhatia, and Tushar Maheshwari. 2018. Aggression-annotated corpus of Hindi-English code-mixed data. arXiv preprint arXiv:1803.09402 (2018).
7. 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. 36–41.
8. Satyajit Kamble and Aditya Joshi. 2018. Hate speech detection from code-mixed Hindi-English tweets using deep learning models. arXiv preprint arXiv:1811.05145 (2018).

References III

9. Krishanu Maity and Sriparna Saha. 2021. BERT-Capsule Model for Cyberbullying Detection in Code-Mixed Indian Languages. In International Conference on Applications of Natural Language to Information Systems. Springer, 147–155.
10. Tulika Saha, Apoorva Upadhyaya, Sriparna Saha, and Pushpak Bhattacharyya. 2021. A Multitask Multimodal Ensemble Model for Sentiment-and Emotion-Aided Tweet Act Classification. IEEE Transactions on Computational Social Systems (2021).
11. Soumitra Ghosh, Asif Ekbal, and Pushpak Bhattacharyya. 2021. A multitask framework to detect depression, sentiment and multi-label emotion from suicide notes. Cognitive Computation (2021), 1–20.
12. Dushyant Singh Chauhan, SR Dhanush, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Sentiment and emotion help sarcasm? a multi-task learning framework for multi-modal sarcasm, sentiment and emotion analysis. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 4351–4360.

References IV

13. Krishanu Maity, Abhishek Kumar, and Sriparna Saha. 2022. A Multi-task Multimodal Framework for Sentiment and Emotion aided Cyberbully Detection. IEEE Internet Computing (2022)..
14. Krishanu Maity and Sriparna Saha. 2021. A Multi-task Model for Sentiment Aided Cyberbullying Detection in Code-Mixed Indian Languages. In International Conference on Neural Information Processing. Springer, 440–451.
15. Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. arXiv preprint arXiv:2005.04790 (2020).
16. Raul Gomez, Jaume Gibert, Lluis Gomez, and Dimosthenis Karatzas. 2020. Exploring hate speech detection in multimodal publications. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 1470–1478

References V

17. Michal Ptaszynski, Fumito Masui, Taisei Nitta, Suzuha Hatakeyama, Yasutomo Kimura, Rafal Rzepka, and Kenji Araki. 2016. Sustainable cyberbullying detection with category-maximized relevance of harmful phrases and double-filtered automatic optimization. International Journal of Child-Computer Interaction 8 (2016), 15–30.
18. Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md Akhtar, Preslav Nakov, Tanmoy Chakraborty, et al. 2021. Detecting harmful memes and their targets. arXiv preprint arXiv:2110.00413 (2021).