MULTIMODAL SENTIMENT ANALYSIS

A Project Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

Bachelor of Technology

in

Department of Computer Science and Engineering

by

Christo Sojan
(Roll No. 2020BCS0069)
Angati Bala Murali
(Roll No. 2020BCS0096)
Kumar Amitanshu
(Roll No. 2020BCS0184)
Syam Sivadas
(Roll No. 2020BCS0024)

Indian Institute of
Information Technology
Kottayam

to

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
KOTTAYAM-686635, INDIA

November 2023

DECLARATION

We, **Christo Sojan** (**Roll No: 2020BCS0069**), **A.Bala Murali** (**Roll No: 2020BCS0096**), **Kumar Amitanshu** (**Roll No: 2020BCS0184**), **Syam Sivdas** (**Roll No: 2020BCS0024**),hereby declare that, this report entitled **"Multimodal Sentiment Analysis"** submitted to Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology** in **Computer Science** is an original work carried out by me under the supervision of **Dr.Manu Madhavan** and has not formed the basis for the award of any degree or diploma, in this or any other institution or university. We have sincerely tried to uphold the academic ethics and honesty. Whenever an external information or statement or result is used then, that have been duly acknowledged and cited.

Kottayam-686635 **Christo Sojan**
November 2023 **A.Bala Murali**
Kumar Amitanshu
Syam Sivdas

CERTIFICATE

This is to certify that the work contained in this project report entitled **"MULTIMODAL SENTIMENT ANALYSIS"** submitted by **Christo Sojan** (**Roll No: 2020BCS0069**), **A.Bala Murali** (**Roll No: 2020BCS0096**), **Kumar Amitanshu** (**Roll No: 2020BCS0184**), **Syam Sivdas** (**Roll No: 2020BCS0024**) to the Indian Institute of Information Technology Kottayam towards partial requirement of **Bachelor of Technology** in **Department of Computer Science** has been carried out by them under my supervision and that it has not been submitted elsewhere for the award of any degree.

Kottayam-686635 (Dr. Manu Madhavan)
November 2023 Project Supervisor

ABSTRACT

The main aim of the project is to advance multimodal sentiment analysis, particularly focusing on the integration of linguistic, acoustic, and visual signals to understand sentiment expressions. The project addresses the challenge of regional language analysis in India, aiming to create a robust dataset encompassing linguistic diversity and cultural nuances. The literature survey covers various research papers on multimodal sentiment analysis, highlighting the use of deep learning architectures and attention mechanisms to improve accuracy and effectiveness. The project's problem statement emphasizes the need for exploring the adaptability of sentiment analysis models to Indian languages. The scope of the project includes improving accuracy in social media sentiment analysis, understanding customer feedback, and enhancing market expansion. The implementation involves the development of computational models to harness the richness of multiple data modalities. The conclusion outlines the current methodology and proposes future enhancements, such as integrating a dense fusion extraction module and expanding to regional languages.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the field of Natural Language Processing (NLP), Natural Language Understanding (NLU) is critical since it is the foundation that machines use to connect with computing systems and human communication. Natural language understanding (NLU) is essential for interpreting user intent, which enables systems to understand and react to a wide range of linguistic phrases. NLU contributes to the creation of intelligent virtual assistants and adaptable interfaces by improving information retrieval accuracy and enabling individualized user interactions by grasping the subtleties of context. NLU guarantees coherent and context-aware discussions in the era of conversational interfaces and chatbots, enhancing the naturalness and user-friendliness of interactions. Additionally, NLU makes sentiment analysis easier, which helps companies get a sense of what the public and their customers think. It has an impact on accessibility as well because it supports advances in fields like machine translation, document summarization, and multimodal interaction and makes technology more inclusive for people with impairments. Fundamentally, natural language understanding (NLU) is the cornerstone of natural language processing (NLP). It enables machines to process language patterns but also to fully comprehend the meaning and intent of human language, thereby enabling the full potential of natural language interactions across a wide range of applications.

1.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that helps computers to understand, interpret, and generate human-like language. It involves the development of algorithms and models capable of processing and analyzing natural language data, which includes text and speech. NLP aims to bridge the gap between human communication and computer understanding, allowing machines to interact with and comprehend language in a manner similar to humans. Sentiment Analysis is a crucial component of NLP as it addresses the nuanced task of deciphering the emotional tone and subjective information embedded in textual data.

1.2 Sentiment Analysis

Sentiment Analysis, also known as opinion mining, is a Natural Language Processing (NLP) technique that involves determining and extracting the sentiment expressed in a piece of text. The goal is to understand the subjective information conveyed in the text and classify it as positive, negative, or neutral. This analysis enables computers to comprehend and interpret the emotional tone behind words, making it a valuable tool in understanding public opinion, customer feedback, or any textual content with subjective elements.

In essence, Sentiment Analysis employs various computational methods to assess the sentiment of a given text, such as customer reviews, social media posts, or news articles. The process often involves machine learning algorithms that are trained on labeled datasets to recognize patterns and associations between words and sentiments.

1.3 Multimodal Sentiment Analysis

Sentiment analysis, traditionally rooted in textual data, has undergone transformative shifts by integrating multimodal inputs. By harnessing the combined power of linguistic, acoustic, and visual signals, we aim to build a comprehensive understanding of sentiment expressions that goes beyond the constraints of individual modalities. Our research is grounded in the belief that true sentiment lies not only in the words spoken or written but also in the tone, intonation, and non-verbal cues accompanying the expression.

1.4 The Challenge of Regional Language Analysis

While the efficacy of multimodal sentiment analysis is being explored globally, our specific focus lies in the diverse linguistic landscape of regional languages in India. The rich tapestry of languages spoken across the subcontinent presents a unique set of challenges and opportunities. Our investigation delves into creating a robust dataset encompassing the linguistic diversity, cultural nuances, and contextual variations inherent in regional languages.

Chapter 2

Literature Survey

2.1 Literature Review

2.1.1 Dense Fusion Network with Multimodal Residual for Sentiment Classification

This paper is based on multimodal sentiment analysis for social media, which contains both text and visual content. In this paper, the authors designed a deep dense Fusion Network with multimodal residual (DFMR) to learn features of modalities individually and then fused them with other (cross-modal dynamics). It's a Hierarchical model in which DFMR first encodes unimodal features independently with the help of Bidirectional Gated Recurrent Unit (Bi-GRU) to model modality-specific interactions. This paper is based on multimodal sentiment analysis for social media, which contains both text and visual content. In this, we designed a deep dense Fusion Network with multimodal residual (DFMR) to learn features of modalities individually and then fused them with other (cross-modal dynamics). The model follows a hierarchical structure, employing a Bidirectional Gated Recurrent Unit (Bi-GRU) to independently encode unimodal features during the initial phase, capturing modality-specific interactions. Subsequently, it incorporates a dense fusion (DF) layer to facilitate cross-modal interactions by combining feature vectors from two associated modalities. As we stated above it is a hierarchical interactive learning process in which bimodal interactions are first learned based on unimodal dynamics and then trimodal dynamics are achieved based on bimodal dynamics. DFMR adopts a multimodal residual (MR) module to retain the fused features of recent layers, especially the ones extracted recently. Finally, the feature representation achieved by MR can be sent to the sentiment classification module for predictions.As we stated above it is a hierarchical interactive learning process in which bimodal interactions are first learned based on unimodal dynamics and then trimodal dynamics are achieved based on bimodal dynamics. DFMR adopts a multimodal residual (MR) module to retain the fused features of recent layers, especially the ones extracted recently. Finally, the feature representation achieved by MR can be sent to the sentiment classification module for predictions.[3]

2.1.2 Multimodal Sentiment Analysis: Addressing Key Issues and Setting up the Baselines

The research paper explores multimodal sentiment analysis using deep-learning architectures for sentiment classification. It considers text, audio, and visual modalities for understanding emotions in videos. The authors use CNN, 3D-CNN, openSMILE, and bc-LSTM models. The bc-LSTM fusion method outperforms SVM in accuracy, while audio and text modalities play ccrucial roles. The paper emphasizes the need for context, different modalities, and generalizability of multimodal sentiment classifiers. Future work should focus on extracting semantics from visual features and incorporating contextual dependency learning. [5]

2.1.3 Multi-Level Attention Map Network for Multimodal Sentiment Analysis

This research paper presents a Multi-Level Attention Map Network (MAMN) for multimodal sentiment analysis (MSA). The paper addresses the challenges of noise reduction, feature extraction, and correlation capture in MSA tasks. The model utilizes techniques such as attention mechanisms, gated mechanisms, and multi-task learning. Experimental results on three public datasets demonstrate that the MAMN model outperforms existing methods in terms of accuracy and effectiveness for document-based and aspect-based MSA tasks. [6]

2.1.4 Multimodal Sentiment Analysis via RNN variants

They conducted experiments on the CMU-MOSI dataset and show that their approach achieves better sentiment classification accuracy than existing methods on individual modalities and also after fusing the modalities using attention networks.[1]

2.1.5 Fusion-Extraction Network for Multimodal Sentiment Analysis

The document proposes a Fusion-Extraction Network (FENet) for multimodal sentiment analysis. The network utilizes an interactive information fusion mechanism to learn visual-specific textual representations and textual-specific visual representations. It also incorporates an information extraction mechanism to filter redundant parts and extract valid information from the multimodal representations. Experimental results on two public multimodal sentiment datasets show that FENet outperforms existing state-of-the-art methods. [4]

Chapter 3

Problem Definition and Scope of the project

3.1 Problem statement

Multimodal Sentiment Analysis has witnessed significant advancement in recent years, particularly with the development of various computational models aiming to harness the richness of multiple data modalities. However, a wide comparison of these models' effectiveness, especially in the context of non-English languages, remains sparse. India, with its rich linguistic diversity, offers a unique opportunity to explore the application of Multimodal Sentiment Analysis, yet research in Indian languages remains underrepresented in the global Multimodal Sentiment Analysis landscape. While some models have shown promise in English or other widely-researched languages, the adaptability of these models to Indian languages remains largely unexplored, signaling both a gap and an opportunity for refinement to capture the differences of these languages.

3.2 Scope of the project

The multi-modal sentiment analysis project holds a greater real-life significance. It focuses on the development of a sentiment analysis for a non-English language. The project's scope is improving the accuracy and understanding of social media, having a deeper understanding of customer feedback, better ideas generation, and market expansion in Product development. A wider range of user emotion information will enhance customer support and media platforms. This project will also help in understanding cultural customs for researchers.

Chapter 4

Implementation and Results

4.1 Implementation

As shown in Figure 4.1, the DFMR structure has four components, namely, the Modality-specific Module, Dense Multimodal Fusion Module, Multimodal Residual Module, and the Sentiment Classification Module. A multimodal sequential data sample, containing language, acoustic, and visual modality, is inputted into the system. The modality-specific component extracts the features of each modality separately. Then, the dense multimodal fusion component combines the multimodal information from the modality-specific features with dense fusion (DF) blocks, and produces fused multimodal features. At the same time, the fused features are fed into the multimodal residual (MR) component to generate multimodal residual features for sentiment classification.[1]

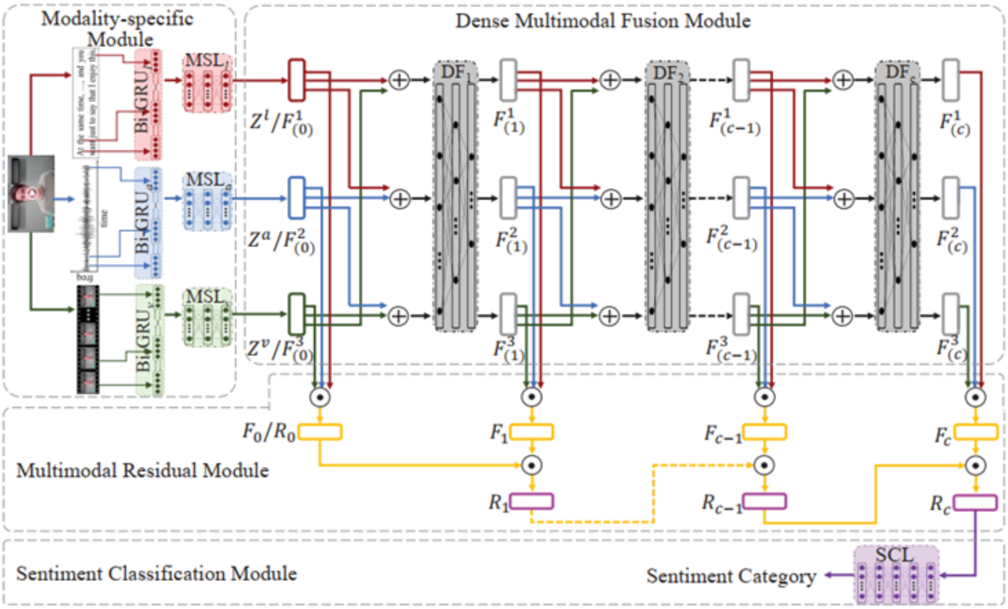


Figure 4.1: Dense Fusion with Multimodal Residual
[3]

- Modality-specific Module: A module that extracts modality-specific features from each sequential modality using bi-directional gated recurrent units (Bi-GRUs) and modality-specific layers (MSLs).

- Dense Multimodal Fusion Module: A module that fuses the modality-specific features in a hierarchical manner using dense fusion (DF) blocks. Each DF block fuses two paired modalities and integrates the fused information with the other modalities.

- Multimodal Residual Module: A module that integrates the modality-specific features and fused features in each DF block using multimodal residual (MR) blocks. Each MR block adds the fused features of the current block and the residual features of the previous block.

- Sentiment Classification Module: A module that predicts the sentiment category of the multimodal data using the final residual feature and sentiment classification layers (SCLs). The objective function is the difference between the true label and the predicted result. Even though we are trying to implement the above architecture, our current architecture is as shown below

4.1.1 Dataset

We have used the extracted features from IEMOCAP dataset for our implementation.[7]

The Interactive Emotional Dyadic Motion Capture (IEMOCAP) database is a multimodal, multi-speaker database of recorded dialogues. The dataset contains 151 videos of 302 recorded dialogues with two speakers per session. Each segment is annotated for nine emotions, including angry, excited, fearful, sad, surprised, frustrated, and happy. The dataset also contains approximately 12 hours of audiovisual data, including video, speech, motion capture of face, and text transcriptions.[2]

4.1.2 Training and testing sets

- trainVid (DICT dialogueId to UtteranceID List): all dialogue IDs for trainset (total: 120)

- testVid (LIST UtteranceID Set): all dialogue IDs for test set (Total: 31)

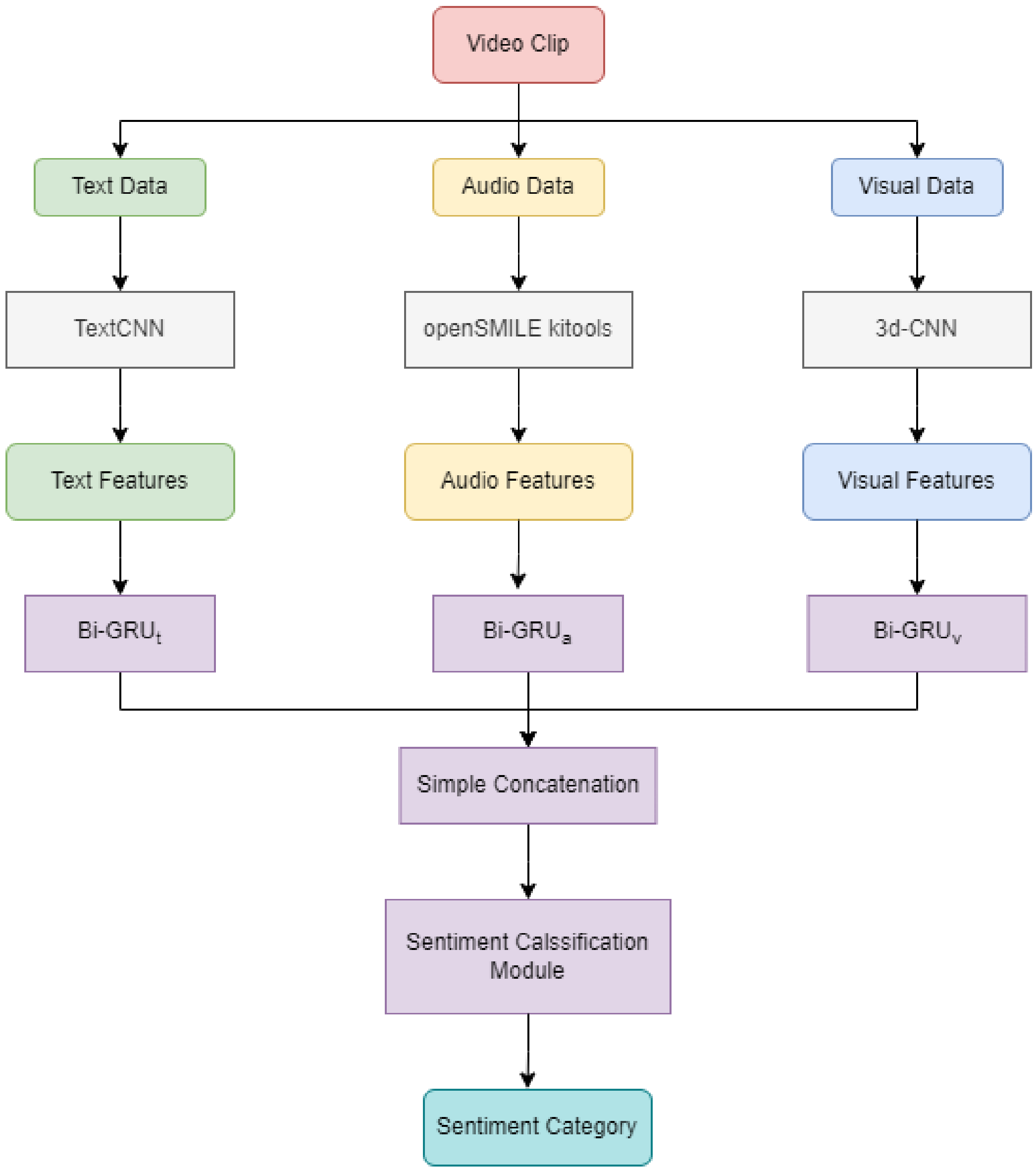


Figure 4.2: Implementation Architecture

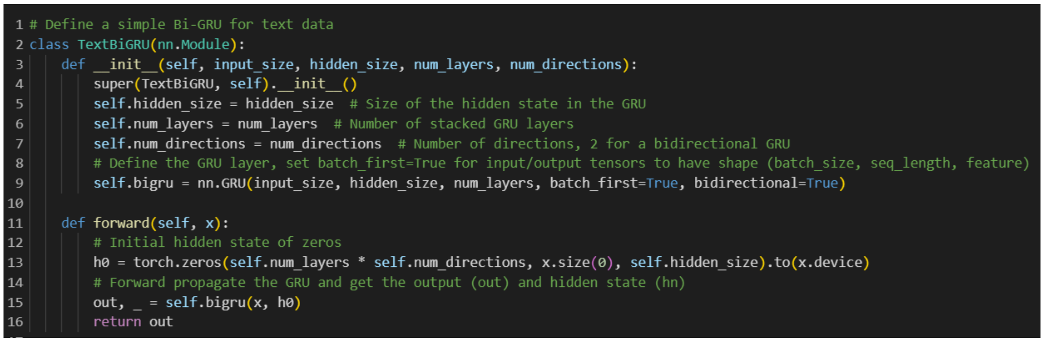
- videoIDs (LIST UtteranceID Set): all dialogue IDs for the whole dataset (total: 120 + 31)

4.1.3 Dataset Content

- videoSpeakers: There are multiple participators in one dialogue. videoSpeakers maps utterance to its speakers
- videoLabels: The emotion Labels for each utterance in a dialogue.
- videoText: The text features extracts using TextCNN.
- videoAudio: The video features extracts using openSMILE kitools
- videoVisual: The visual features extracts using 3d-CNN.
- videoSentence: The raw text info in a dialogue.

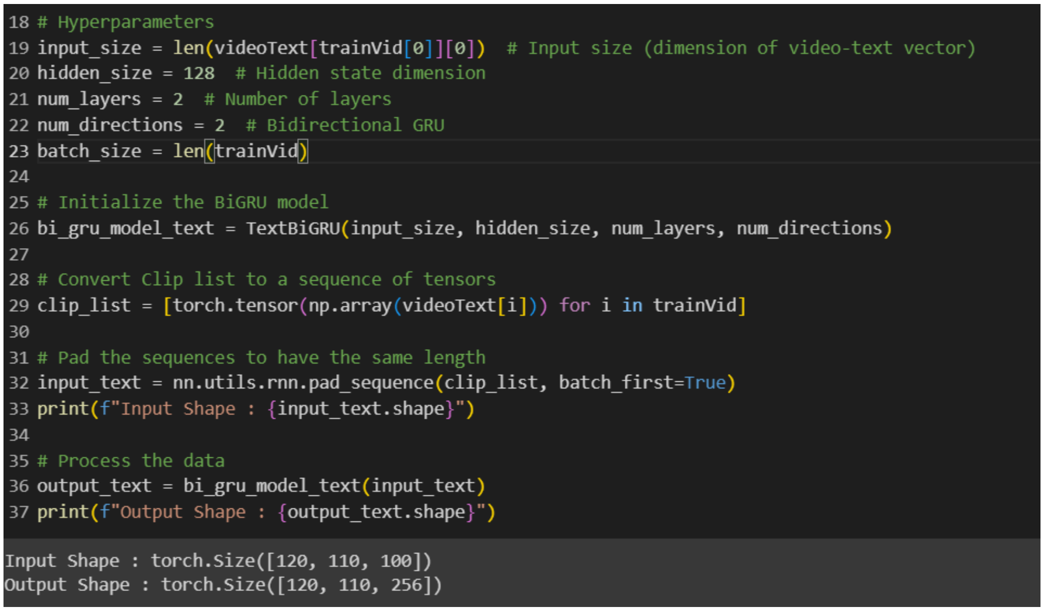
4.1.4 Extracting and Combining Features

- Text Feature Processing: A Bi-GRU model is defined for processing text features. Text features are converted into a sequence of tensors, padded for uniform length, and then passed through the Bi-GRU model. Definition of Bi-GRU for Text Modality is shown in Figure 4.3 and Figure 4.4.



```python
# Define a simple Bi-GRU for text data
class TextBiGRU(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, num_directions):
        super(TextBiGRU, self).__init__()
        self.hidden_size = hidden_size  # Size of the hidden state in the GRU
        self.num_layers = num_layers  # Number of stacked GRU layers
        self.num_directions = num_directions  # Number of directions, 2 for a bidirectional GRU
        # Define the GRU layer, set batch_first=True for input/output tensors to have shape (batch_size, seq_length, feature)
        self.bigru = nn.GRU(input_size, hidden_size, num_layers, batch_first=True, bidirectional=True)

    def forward(self, x):
        # Initial hidden state of zeros
        h0 = torch.zeros(self.num_layers * self.num_directions, x.size(0), self.hidden_size).to(x.device)
        # Forward propagate the GRU and get the output (out) and hidden state (hn)
        out, _ = self.bigru(x, h0)
        return out
```

Figure 4.3: Text Bi-GRU



```
# Hyperparameters
input_size = len(videoText[trainVid[0]][0])  # Input size (dimension of video-text vector)
hidden_size = 128  # Hidden state dimension
num_layers = 2  # Number of layers
num_directions = 2  # Bidirectional GRU
batch_size = len(trainVid)

# Initialize the BiGRU model
bi_gru_model_text = TextBiGRU(input_size, hidden_size, num_layers, num_directions)

# Convert Clip list to a sequence of tensors
clip_list = [torch.tensor(np.array(videoText[i])) for i in trainVid]

# Pad the sequences to have the same length
input_text = nn.utils.rnn.pad_sequence(clip_list, batch_first=True)
print(f"Input Shape : {input_text.shape}")

# Process the data
output_text = bi_gru_model_text(input_text)
print(f"Output Shape : {output_text.shape}")
```

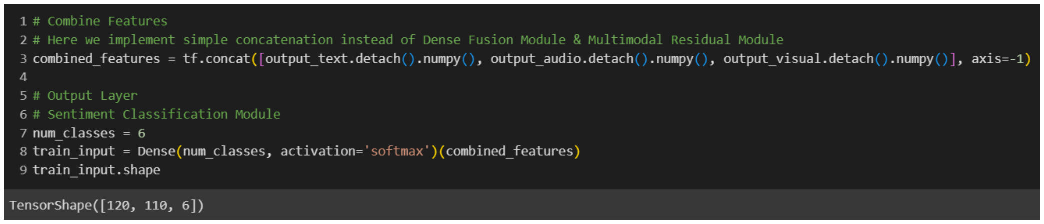
```
Input Shape : torch.Size([120, 110, 100])
Output Shape : torch.Size([120, 110, 256])
```

Figure 4.4: Text Bi-GRU Initialization

- Audio Feature Processing: Similar to text, a Bi-GRU model is defined for processing audio features. Audio features are processed in the same manner as text features.

- Visual Feature Processing: Similar to text and audio, a Bi-GRU model is defined for processing visual features. Visual features are processed in the same manner as text and audio features.

- Combining Features: The outputs from the Bi-GRU models for text, audio, and visual features are concatenated to create a combined feature representation. A simple Dense layer is added to the concatenated features for sentiment classification. It is shown in Figure 4.5.



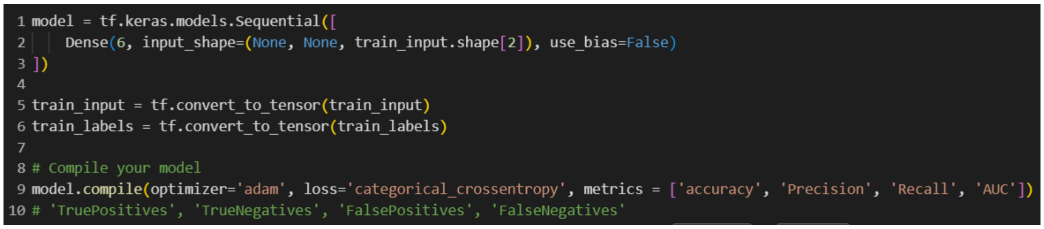
```
# Combine Features
# Here we implement simple concatenation instead of Dense Fusion Module & Multimodal Residual Module
combined_features = tf.concat([output_text.detach().numpy(), output_audio.detach().numpy(), output_visual.detach().numpy()], axis=-1)

# Output Layer
# Sentiment Classification Module
num_classes = 6
train_input = Dense(num_classes, activation='softmax')(combined_features)
train_input.shape
```

```
TensorShape([120, 110, 6])
```

Figure 4.5: Combining Features

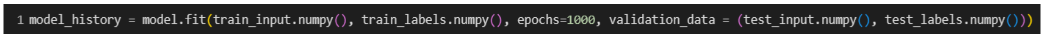
4.1.5 Model Construction, Compilation, Training and Evaluation

The model is compiled using the Adam optimizer and categorical cross entropy loss. Labels are transformed into a suitable format for training. The model is trained using the training data with validation on the test data. It is shown in Figure 4.6 and Figure 4.7.



```
model = tf.keras.models.Sequential([
    Dense(6, input_shape=(None, None, train_input.shape[2]), use_bias=False)
])

train_input = tf.convert_to_tensor(train_input)
train_labels = tf.convert_to_tensor(train_labels)

# Compile your model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics = ['accuracy', 'Precision', 'Recall', 'AUC'])
# 'TruePositives', 'TrueNegatives', 'FalsePositives', 'FalseNegatives'
```

Figure 4.6: Model Construction and Compilation



```
model_history = model.fit(train_input.numpy(), train_labels.numpy(), epochs=1000, validation_data = (test_input.numpy(), test_labels.numpy()))
```

Figure 4.7: Model Training and Evaluation

4.2 Results

	Text	**Audio**	**Visual**	**MSA**
Training Model Accuracy	0.1652	0.0764	0.0717	0.5384
Testing Model Accuracy	0.1641	0.0705	0.0599	0.1648
Training Model Loss	2.7573	3.1480	4.0246	1.8947
Testing Model Loss	3.8033	4.2748	5.3365	3.8968

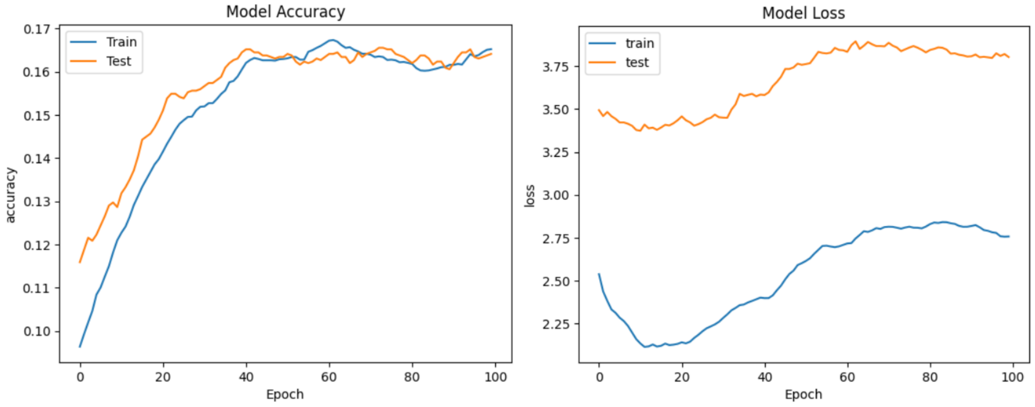
Table 4.1: Comparison of Different Modalities

The table, shown in Figure 4.1, represents the performance of a machine learning model trained on four different types of data: Text, Audio, Visual, and MSA (Multimodal sentiment Analysis). The performance is evaluated in terms of accuracy and loss for both training and testing datasets.

Training Model Accuracy: This row shows the accuracy of the model on the training dataset. Accuracy is a measure of how often the model's predictions match the actual labels. The values range from 0 to 1, with 1 indicating perfect accuracy. In this case, the model performs best on MSA data with an accuracy of 0.5384, and worst on Audio data with an accuracy of 0.0764.

Testing Model Accuracy: This row shows the accuracy of the model on the testing dataset. This is a measure of how well the model generalizes to unseen data. Again, the model performs best on MSA data with an accuracy of 0.1648, and worst on Visual data with an accuracy of 0.0599.

Training Model Loss: This row shows the loss of the model on the training dataset. Loss is a measure of the error made by the model's predictions. A lower loss indicates a better model. The values for this row are not fully visible in the image, but from what can be seen, the model has a higher loss on Audio data (3.148) compared to Text data (2.7573).



(a) Model Accuracy for Text Modality (b) Model Loss for Text Modality

Figure 4.8: Text Modality

The graph, shown in Figure 4.8 (a), is a line graph that represents the accuracy of a text model over time. The x-axis represents the number of epochs, and the y-axis represents the accuracy.

There are two lines on the graph:

The blue line represents the accuracy of the model on the training data. This line starts at around 0.11 and increases steadily to around 0.16 at epoch 100, which indicates that the model is learning and improving its performance on the training data.

The orange line represents the accuracy of the model on the test data. This line starts at around 0.13 and increases steadily to around 0.17 at epoch 100, which indicates that the model is also improving its performance on the test data.

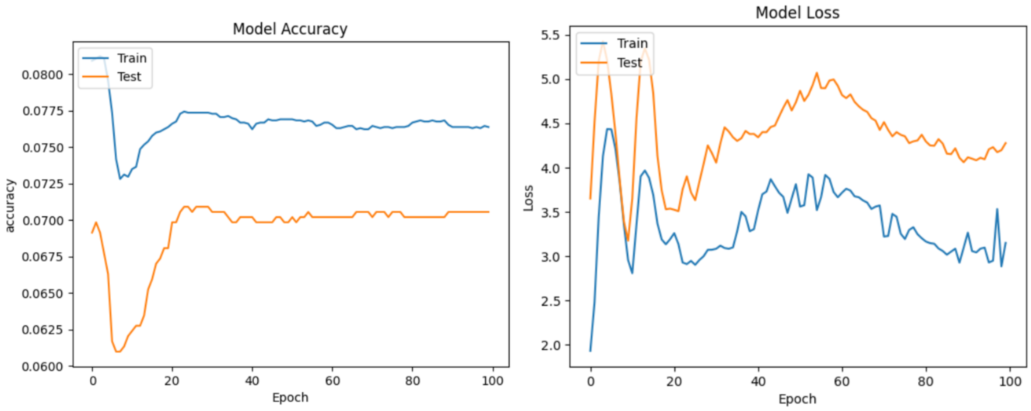
The fact that both the training and test accuracies are increasing might suggest that the model is learning effectively from the training data and generalizing well to the test data.

The graph, as shown in Figure 4.8 (b), is a line graph that represents the loss of a text model over time. The x-axis represents the number of epochs, and the y-axis represents the loss.

There are two lines on the graph:

The blue line represents the loss of the model on the training data. This line has a steep decrease in loss at the beginning of the epochs and then levels off, which indicates that the model is learning and improving its performance on the training data.

The orange line represents the loss of the model on the test data. This line has a higher loss than the blue line for most of the epochs, which indicates that the model's performance on the test data is not as good as on the training data.



(a) Model Accuracy for Audio Modality (b) Model Loss for Audio Modality

Figure 4.9: Audio Modality

The graph, as shown in Figure 4.9 (a), is a line graph that represents the accuracy of a model for audio feature analysis . The x-axis represents the number of epochs, and the y-axis represents the accuracy.

There are two lines on the graph:

The blue line represents the accuracy of the model on the training data. This line is higher, which indicates that the model is performing well on the training data.

The orange line represents the accuracy of the model on the test data. This line is lower, which indicates that the model is not performing as well on the test data as it is on the training data.

The fact that the training data line is consistently higher than the test data line suggests that the model may be overfitting.

The graph, as shown in Figure 4.9 (b), is a line graph that represents the loss of a model over time. The x-axis represents the number of epochs, and the y-axis represents the loss.

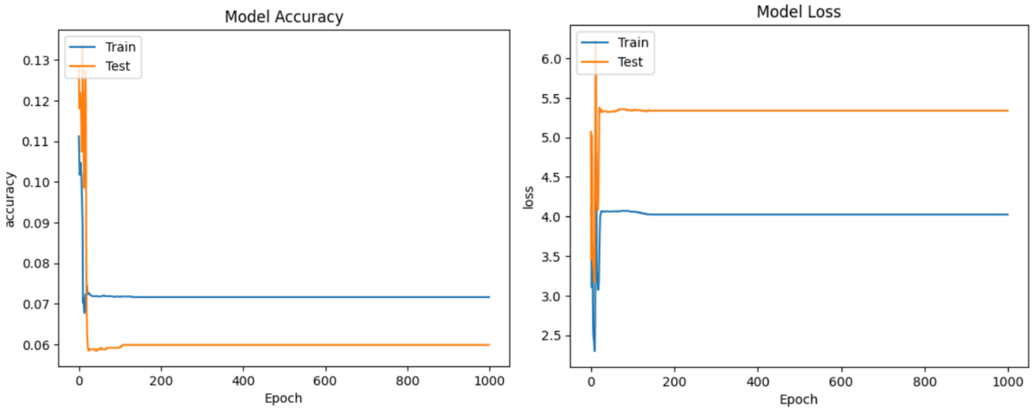
There are two lines on the graph:

The blue line represents the loss of the model on the training data. This line decreases over time, which indicates that the model is learning and improving its performance on the training data. The orange line represents the loss of the model on the test data. This line fluctuates, which indicates that the model's performance on the test data is not stable.

The graph, as shown in Figure 4.10 (a), is a line graph that represents the accuracy of a visual model over time. The x-axis represents the number of epochs, and the y-axis represents the accuracy.

There are two lines on the graph:

The blue line represents the accuracy of the model on the training data. This line shows that the accuracy of the model during training increases



(a) Model Accuracy for Visual Modality (b) Model Loss for Visual Modality

Figure 4.10: Visual Modality

rapidly at first and then levels off, which indicates that the model is learning and improving its performance on the training data.

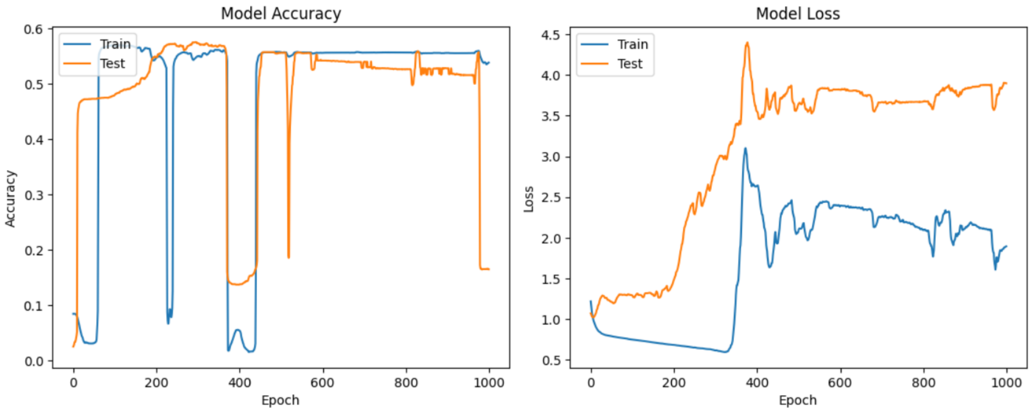
The orange line represents the accuracy of the model on the test data. This line shows that the accuracy of the model during testing remains relatively constant throughout, which indicates that the model's performance on the test data is stable.

The graph, as shown in Figure 4.10 (b), is a line graph that represents the loss of a visual model over time. The x-axis represents the number of epochs, and the y-axis represents the loss.

There are two lines on the graph:

The blue line represents the loss of the model on the training data. This line shows that the loss of the model during training decreases as the number of epochs increases, which indicates that the model is learning and improving its performance on the training data.

The orange line represents the loss of the model on the test data. This line



(a) Model Accuracy for Multimodal Sentiment Analysis (b) Model Loss for Multimodal Sentiment Analysis

Figure 4.11: Multimodal Sentiment Analysis

shows that the loss of the model during testing remains relatively constant throughout, which indicates that the model's performance on the test data is stable.

The graph, as shown in Figure 4.11 (a), represents the accuracy of a machine learning model during its training and testing phases. Here's a breakdown of the graph:

The x-axis represents the number of epochs. An epoch is one complete pass through the entire training dataset. As you move right along the x-axis, the number of epochs increases.

The y-axis represents the accuracy of the model. Accuracy is a measure of how often the model's predictions are correct. As you move up along the y-axis, the accuracy of the model increases.

The blue line represents the training accuracy. This is how well the model performs on the same data it was trained on. As the number of epochs increases, the training accuracy also increases, indicating that the model is learning from the training data.

The orange line represents the testing accuracy. This is how well the model performs on new, unseen data. The testing accuracy increases with the number of epochs initially, but it starts to plateau after around 600 epochs. This could be an indication that the model is starting to overfit the training data.

The line graph, as shown in Figure 4.11 (b), represents the model's loss over a series of epochs for both the training and testing datasets.

Here's a brief explanation:

The x-axis represents the number of epochs. An epoch is one complete pass through the entire training dataset. In this graph, it appears to span over 1000 epochs.

The y-axis represents the loss. The loss is a measure of how well the model's predictions match the actual labels. A lower loss indicates a better model.

The blue line represents the loss for the training dataset. This line shows a decreasing trend, indicating that the model is learning and improving its performance on the training data over time.

The orange line represents the loss for the testing dataset. This line shows an increasing trend, suggesting that the model's performance on the testing data is deteriorating over time. This could be a sign of overfitting, where the model performs well on the training data but poorly on unseen data.

Chapter 5

Conclusion and Future works

In the pursuit of advancing multimodal sentiment analysis, our current efforts are constrained by resource limitations, necessitating the use of a smaller dataset. However, a strategic plan is in place to transition to the larger CMU MOSEI dataset, with a focus on optimizing computational efficiency. This report outlines our current methodology, which involves simple concatenation of modalities, and proposes the integration of a dense fusion extraction module to enhance accuracy.

To elevate the accuracy of our multimodal sentiment analysis, we will be replacing the existing simple concatenation approach with a dense fusion extraction module. This module, potentially leveraging attention mechanisms or transformer-based architectures, aims to capture intricate relationships between modalities. Rigorous training and validation procedures will be implemented to fine-tune and optimize its parameters.

The expansion of our model to include regional languages in India requires meticulous dataset creation, involving collaboration with linguistic experts to ensure linguistic accuracy. Maintaining consistent accuracy across languages will involve language-specific fine-tuning, preprocessing, and feature engineering. Cultural nuances in sentiment expression will be considered during model development.

Ensuring consistent accuracy across regional languages necessitates ongoing evaluation and adaptation. Fine-tuning the model on each language separately, incorporating language-specific features or embeddings, and staying attuned to language-specific challenges are integral aspects of achieving this objective.

In conclusion, our current efforts and enhancements mark a substantial stride toward advancing multimodal sentiment analysis. The integration of the dense fusion extraction module and the expansion to regional languages underscore our commitment to both precision and inclusivity in sentiment analysis.

Bibliography

[1] Ayush Agarwal, Ashima Yadav, and Dinesh Kumar Vishwakarma. Multimodal sentiment analysis via rnn variants. In *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science Engineering (BCD)*, pages 19–23, 2019.

[2] C. Lee A. Kazemzadeh E. Mower S. Kim J. Chang S. Lee C. Busso, M. Bulut and S. Narayanan. Iemocap: Interactive emotional dyadic motion capture database, 2008. https://sail.usc.edu/iemocap/index.html.

[3] Huan Deng, Peipei Kang, Zhenguo Yang, Tianyong Hao, Qing Li, and Wenyin Liu. Dense fusion network with multimodal residual for sentiment classification. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021.

[4] Tao Jiang, Jiahai Wang, Zhiyue Liu, and Yingbiao Ling. Fusion-extraction network for multimodal sentiment analysis. In Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan, editors, *Advances in Knowledge Discovery and Data Mining*, pages 785–797, Cham, 2020. Springer International Publishing.

[5] Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Erik Cambria, Alexander Gelbukh, and Amir Hussain. Multimodal sentiment analysis: Addressing key issues and setting up the baselines. *IEEE Intelligent Systems*, 33(6):17–25, 2018.

[6] Xiaojun Xue, Chunxia Zhang, Zhendong Niu, and Xindong Wu. Multi-level attention map network for multimodal sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(5):5105–5118, 2023.

[7] Ziqi Yuan. Iemocap, 2020. https://www.kaggle.com/datasets/columbine/iemocap.