Multimodal Emotion Recognition (Audio/Visual/Text)

Salika Akeek Dave
sad6003@psu.edu

1. Task

The task is to develop a system for recognizing emotions from audio, visual, and text data. This involves processing multimodal inputs to identify the emotional content, which can be a challenging task due to the complexity of interpreting and integrating information from different modalities. The project aims to address this challenge by employing advanced techniques to extract and fuse information from multiple modalities, ultimately enabling the accurate recognition of emotions across various data sources. The dataset in focus across which multiple works have been compared is the *The Interactive Emotional Dyadic Motion Capture (IEMOCAP) Database* released by USC Viterbi School of Engineering [1]

2. Related Work

The first paper exploring this task on this dataset is as described in *Multimodal Sentiment Analysis using Hierarchical Fusion with Context Modeling* [2]. The technical approach taken in the paper addresses the challenge of fusing features from different modalities in multimodal sentiment analysis. The common practice in this field has been to simply concatenate the feature vectors of different modalities. However, this method does not account for the possibility that different modalities may carry conflicting information. It introduces an innovative feature fusion strategy that improves the multimodal fusion mechanism. This strategy hierarchically fuses modalities, first two by two, and then all three together. The paper employs a recurrent neural network (RNN) to propagate contextual information between utterances in a video clip. This approach aims to enhance the accuracy and effectiveness of sentiment analysis by considering the context provided by the sequence of utterances in a video

Previous models as described in *Combining Deep and Unsupervised Features for Multilingual Speech Emotion Recognition* [3] recognize emotions from spoken sentences in multiple languages by using a Convolutional Neural Network (CNN) with an end-to-end deep architecture. This model takes raw text and audio data and utilizes convolutional layers to extract a hierarchy of classification features. Additionally, the model achieves good performances in different languages thanks to the usage of multilingual unsupervised textual features. Importantly, the proposed model, PATHOSnet, does not require text and audio to be word- or phoneme-aligned, making it compatible with multiple languages

The SOTA for the given task can be considered to be "COGMEN" [4] which explores the modeling of intra-speaker dependency between utterances in a conversation using a Relational Graph Convolutional Network (RGCN) and a Graph Transformer. Each utterance is represented as a node, and the paper focuses on constructing graphs with relations to capture inter-speaker and intra-speaker dependency on the connected utterances. The paper also explores the effect of window size in the Graph Formation module of the architecture, treating it as a hyperparameter that can be adjusted during training. The window size is found to impact the performance based on the maintenance of inter and intra-speaker dependencies for different sequence lengths. Please refer to section on the approach for more details.
The source of truth for comparison of these models is as described by papers with code [6] on the IEMOCAP dataset [1].

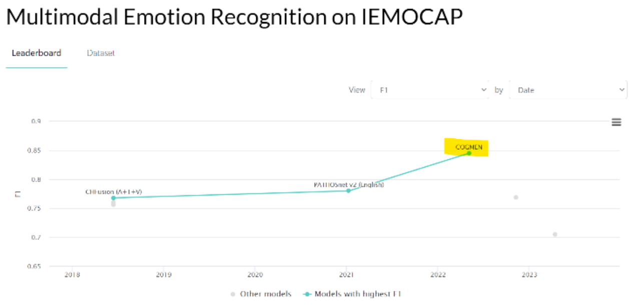


Figure 1. SOTA – COGMEN shows the highest F1 Score

3. Approach

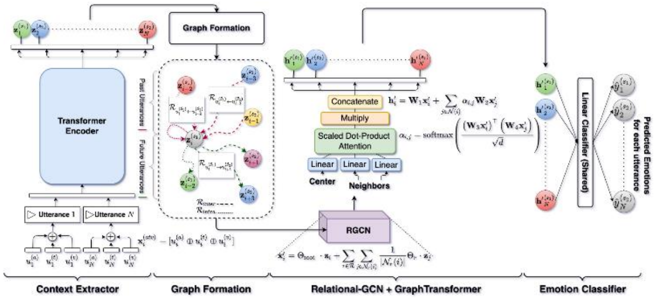

Graph Formation
Transformer Encoder
Past Utterances
Future Utterances
Utterance 1
Utterance N
$x_i^{(atv)} = [u_i^{(a)} \oplus u_i^{(t)} \oplus u_i^{(v)}]$
$\mathcal{R}_{inter}$
$\mathcal{R}_{intra}$
Concatenate
Multiply
Scaled Dot-Product Attention
Linear
Linear
Linear
Center
Neighbors
RGCN
$h_i' = W_1 x_i' + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} W_2 x_j'$
$\alpha_{i,j} = \text{softmax}\left(\frac{(W_3 x_i')^\top (W_4 x_j')}{\sqrt{d}}\right)$
$x_i' = \Theta_{root} \cdot x_i + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|} \Theta_r \cdot x_j$
Linear Classifier (Shared)
Predicted Emotions for each utterance
Context Extractor
Graph Formation
Relational-GCN + GraphTransformer
Emotion Classifier

Figure 2. COGMEN Model Architecture. Figure derived from COGMEN [4]

Key Libraries used by COGMEN [4]:

1. PyG (PyTorch Geometric) for the GNN component [7]
2. Comet for logging all experiments and its Bayesian optimizer for hyperparameter tuning. [8]
3. SentenceBERT for textual features. [9]

4. Dataset

The Interactive Emotional Dyadic Motion Capture (IEMOCAP) [1] dataset is a widely used multimodal dataset in the field of affective computing and emotion recognition. It was developed by the Speech Analysis and Interpretation Laboratory (SAIL) at the University of Southern California (USC). It contains approximately 12 hours of audiovisual data, including video, speech, motion capture of face, text transcriptions. It consists of dyadic sessions where actors perform improvisations or scripted scenarios, specifically selected to elicit emotional expressions. IEMOCAP database is annotated by multiple annotators into categorical labels, such as anger, happiness, sadness, neutrality, as well as dimensional labels such as valence, activation and dominance. The detailed motion capture information, the interactive setting to elicit authentic emotions, and the size of the database make this corpus a valuable addition to the existing databases in the community for the study and modeling of multimodal and expressive human communication. Key features of the IEMOCAP dataset include:

1. Multimodal Data: The dataset captures a variety of modalities, including audio, video, and text. This allows researchers to explore the emotional content expressed through different channels.

2. Dyadic Interactions: The dataset consists of recordings of dyadic interactions, where two actors engage in scripted scenarios designed to elicit specific emotional responses. This design allows for the study of how emotions are expressed and perceived in interactive settings.

3. Actors and Scenarios: The dataset involves ten actors (five male and five female) and contains a total of approximately 12 hours of audiovisual recordings. The actors perform in various scenarios, ranging from casual conversations to more emotionally charged situations.

4. Emotion Annotations: Emotion labels are provided for each session in the dataset. These labels are typically based on the consensus of human annotators who watch and annotate the recordings. Emotions commonly labeled include happiness, sadness, anger, frustration, excitement, and more.

5. Phonetic Transcriptions: The dataset also includes phonetic transcriptions of the spoken content, allowing researchers to analyze the speech patterns associated with different emotions.

6. Motion Capture Data: In addition to audio and video recordings, the IEMOCAP dataset includes motion capture data, capturing the facial expressions and body movements of the actors during the interactions.

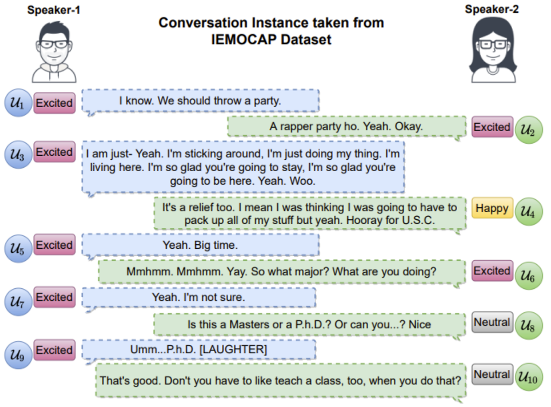
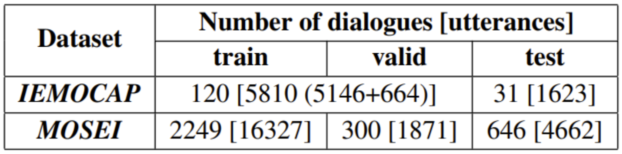

Speaker-1
Conversation Instance taken from IEMOCAP Dataset
Speaker-2
$\mathcal{U}_1$ Excited
I know. We should throw a party.
A rapper party ho. Yeah. Okay.
Excited $\mathcal{U}_2$
$\mathcal{U}_3$ Excited
I am just- Yeah. I'm sticking around, I'm just doing my thing. I'm living here. I'm so glad you're going to stay, I'm so glad you're going to be here. Yeah. Woo.
It's a relief too. I mean I was thinking I was going to have to pack up all of my stuff but yeah. Hooray for U.S.C.
Happy $\mathcal{U}_4$
$\mathcal{U}_5$ Excited
Yeah. Big time.
Mmhmm. Mmhmm. Yay. So what major? What are you doing?
Excited $\mathcal{U}_6$
$\mathcal{U}_7$ Excited
Yeah. I'm not sure.
Is this a Masters or a P.h.D.? Or can you...? Nice
Neutral $\mathcal{U}_8$
$\mathcal{U}_9$ Excited
Umm...P.h.D. [LAUGHTER]
That's good. Don't you have to like teach a class, too, when you do that?
Neutral $\mathcal{U}_{10}$

Figure 3. An example conversation between two speakers, with corresponding emotions evoked for each utterance. Figure derived from COGMEN [4]



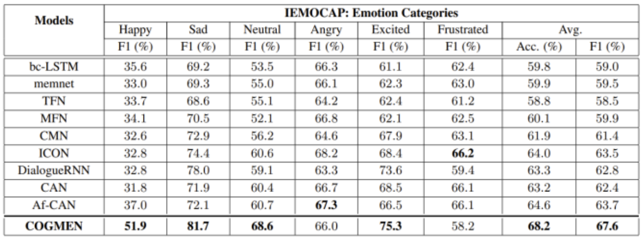
Dataset	Number of dialogues [utterances]		
	train	valid	test
IEMOCAP	120 [5810 (5146+664)]		31 [1623]
MOSEI	2249 [16327]	300 [1871]	646 [4662]

Figure 4. Dataset Statistics. Figure derived from COGMEN [4]

However, this dataset requires a procedure for obtaining special access. For the purpose of replicating results, the authors provided a pickle file of the data and features in the official PyTorch implementation of COGMEN [5].

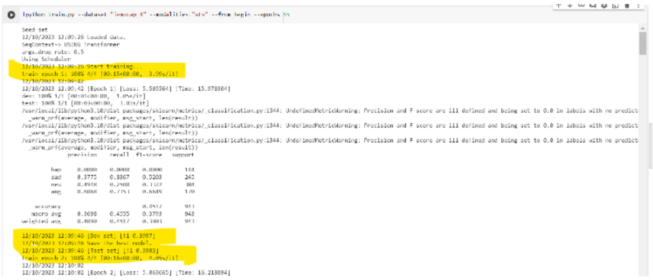
5. Results

With the help of the PyTorch implementation of COGMEN [5] I have been able to reproduce the results of the paper and have mainly used the metrics – Precision, Recall, F1 Score to compare the results. I have retrained the model for 4-way test set.



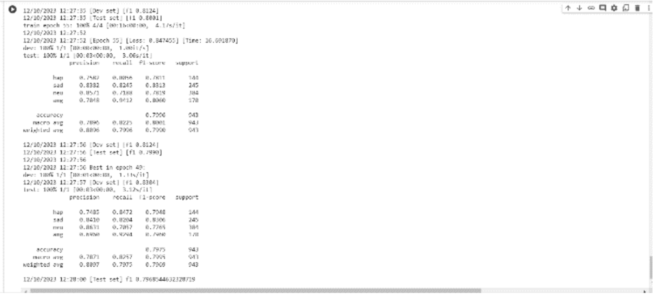
Models	IEMOCAP: Emotion Categories							
	Happy	Sad	Neutral	Angry	Excited	Frustrated	Avg.	
	F1 (%)	F1 (%)	F1 (%)	F1 (%)	F1 (%)	F1 (%)	Acc. (%)	F1 (%)
bc-LSTM	35.6	69.2	53.5	66.3	61.1	62.4	59.8	59.0
memnet	33.0	69.3	55.0	66.1	62.3	63.0	59.9	59.5
TFN	33.7	68.6	55.1	64.2	62.4	61.2	58.8	58.5
MFN	34.1	70.5	52.1	66.8	62.1	62.5	60.1	59.9
CMN	32.6	72.9	56.2	64.6	67.9	63.1	61.9	61.4
ICON	32.8	74.4	60.6	68.2	68.4	**66.2**	64.0	63.5
DialogueRNN	32.8	78.0	59.1	63.3	73.6	59.4	63.3	62.8
CAN	31.8	71.9	60.4	66.7	68.5	66.1	63.2	62.4
Af-CAN	37.0	72.1	60.7	**67.3**	66.5	66.1	64.6	63.7
COGMEN	**51.9**	**81.7**	**68.6**	66.0	**75.3**	58.2	**68.2**	**67.6**

Figure 5. Results reported by the paper for a 6-way test set.



```
!python train.py --dataset="iemocap_4" --modalities="atv" --from_begin --epochs 55
Seed set
12/10/2023 12:09:26 Loaded data.
SeqContext-> USING Transformer
args.drop_rate: 0.5
Using Scheduler
12/10/2023 12:09:26 Start training...
train epoch 1: 100% 4/4 [00:15<00:00,  3.99s/it]
12/10/2023 12:09:42
12/10/2023 12:09:42 [Epoch 1] [Loss: 5.505564] [Time: 15.078304]
dev: 100% 1/1 [00:01<00:00,  1.05s/it]
test: 100% 1/1 [00:03<00:00,  3.03s/it]
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predict
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predict
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predict
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

         hap     0.0000    0.0000    0.0000       144
         sad     0.3775    0.8367    0.5203       245
         neu     0.4908    0.2500    0.3322       384
         ang     0.6068    0.7151    0.6649       170

    accuracy                         0.4517       943
   macro avg     0.3698    0.4355    0.3793       943
weighted avg     0.4890    0.4517    0.3981       943

12/10/2023 12:09:46 [Dev set] [f1 0.3997]
12/10/2023 12:09:46 Save the best model.
12/10/2023 12:09:46 [Test set] [f1 0.3903]
train epoch 2: 100% 4/4 [00:16<00:00,  4.06s/it]
12/10/2023 12:10:02
12/10/2023 12:10:02 [Epoch 2] [Loss: 5.063665] [Time: 16.218894]
```

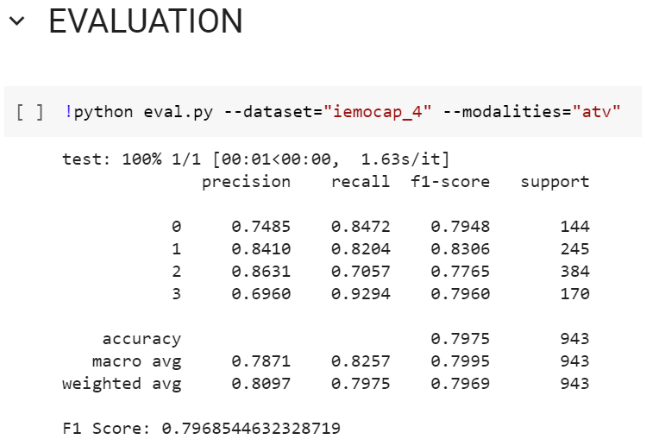
Figure 6. Epoch 1: F1 Score: 0.399 [dev set] and 0.3903 [test set].



```
12/10/2023 12:27:35 [Dev set] [f1 0.8124]
12/10/2023 12:27:35 [Test set] [f1 0.8081]
train epoch 55: 100% 4/4 [00:16<00:00,  4.17s/it]
12/10/2023 12:27:52
12/10/2023 12:27:52 [Epoch 55] [Loss: 0.847455] [Time: 16.691870]
dev: 100% 1/1 [00:00<00:00,  1.00it/s]
test: 100% 1/1 [00:03<00:00,  3.06s/it]
              precision    recall  f1-score   support

         hap     0.7582    0.8056    0.7811       144
         sad     0.8382    0.8245    0.8313       245
         neu     0.8571    0.7188    0.7819       384
         ang     0.7048    0.9412    0.8060       170

    accuracy                         0.7996       943
   macro avg     0.7896    0.8225    0.8001       943
weighted avg     0.8096    0.7996    0.7990       943

12/10/2023 12:27:56 [Dev set] [f1 0.8124]
12/10/2023 12:27:56 [Test set] [f1 0.7990]
12/10/2023 12:27:56
12/10/2023 12:27:56 Best in epoch 49:
dev: 100% 1/1 [00:01<00:00,  1.11s/it]
12/10/2023 12:27:57 [Dev set] [f1 0.8304]
test: 100% 1/1 [00:03<00:00,  3.12s/it]
              precision    recall  f1-score   support

         hap     0.7485    0.8472    0.7948       144
         sad     0.8410    0.8204    0.8306       245
         neu     0.8631    0.7057    0.7765       384
         ang     0.6960    0.9294    0.7960       170

    accuracy                         0.7975       943
   macro avg     0.7871    0.8257    0.7995       943
weighted avg     0.8097    0.7975    0.7969       943

12/10/2023 12:28:00 [Test set] f1 0.7968544632328719
```

Figure 7. Epoch 55: F1 Score: 0.8124 [dev set] and 0.7990 [test set].

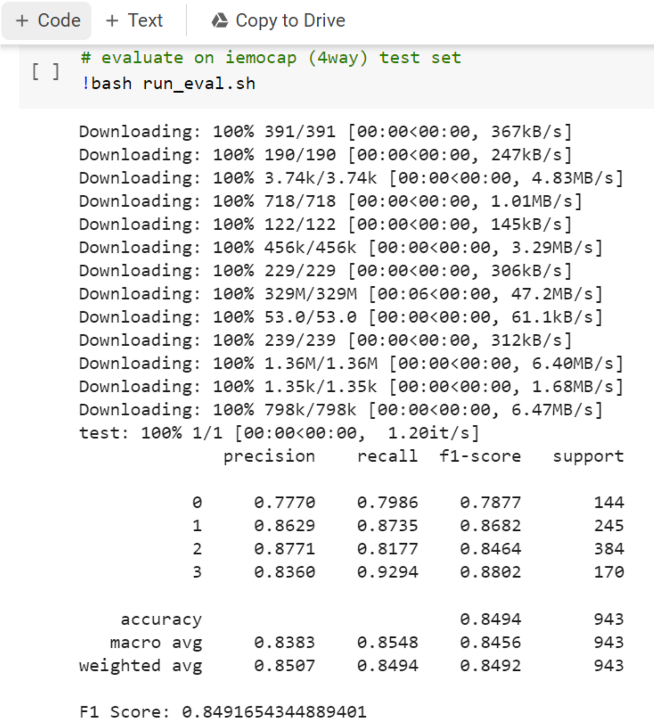
Metrics for all epochs available at https://github.com/salikadave/CSE597CourseProject/blob/main/training_logs.txt .



EVALUATION

```
[ ] !python eval.py --dataset="iemocap_4" --modalities="atv"

test: 100% 1/1 [00:01<00:00,  1.63s/it]
              precision    recall  f1-score   support

           0     0.7485    0.8472    0.7948       144
           1     0.8410    0.8204    0.8306       245
           2     0.8631    0.7057    0.7765       384
           3     0.6960    0.9294    0.7960       170

    accuracy                         0.7975       943
   macro avg     0.7871    0.8257    0.7995       943
weighted avg     0.8097    0.7975    0.7969       943

F1 Score:  0.7968544632328719
```

Figure 7. Running evaluation on the trained model gives F1 Score = 0.79685



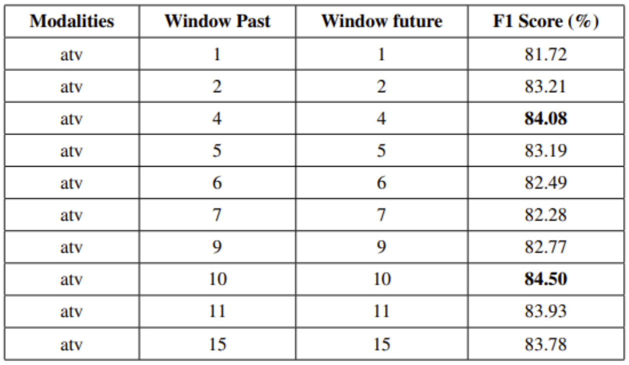
```
[ ] # evaluate on iemocap (4way) test set
    !bash run_eval.sh

Downloading: 100% 391/391 [00:00<00:00, 367kB/s]
Downloading: 100% 190/190 [00:00<00:00, 247kB/s]
Downloading: 100% 3.74k/3.74k [00:00<00:00, 4.83MB/s]
Downloading: 100% 718/718 [00:00<00:00, 1.01MB/s]
Downloading: 100% 122/122 [00:00<00:00, 145kB/s]
Downloading: 100% 456k/456k [00:00<00:00, 3.29MB/s]
Downloading: 100% 229/229 [00:00<00:00, 306kB/s]
Downloading: 100% 329M/329M [00:06<00:00, 47.2MB/s]
Downloading: 100% 53.0/53.0 [00:00<00:00, 61.1kB/s]
Downloading: 100% 239/239 [00:00<00:00, 312kB/s]
Downloading: 100% 1.36M/1.36M [00:00<00:00, 6.40MB/s]
Downloading: 100% 1.35k/1.35k [00:00<00:00, 1.68MB/s]
Downloading: 100% 798k/798k [00:00<00:00, 6.47MB/s]
test: 100% 1/1 [00:00<00:00,  1.20it/s]
              precision    recall  f1-score   support

           0     0.7770    0.7986    0.7877       144
           1     0.8629    0.8735    0.8682       245
           2     0.8771    0.8177    0.8464       384
           3     0.8360    0.9294    0.8802       170

    accuracy                         0.8494       943
   macro avg     0.8383    0.8548    0.8456       943
weighted avg     0.8507    0.8494    0.8492       943

F1 Score:  0.8491654344889401
```

Figure 7. Running evaluation on the available model gives F1 Score = 0.8491

As we can see from figure 6 and 7, the F1 Score is lesser after re-training. As reported by other users as well, a possible reason for a lesser score is the authors making use of the Bayesian Optimizer that is part of the COMET ML library. This is still an open issue and can be found here: https://github.com/Exploration-Lab/COGMEN/issues/1

6. Possible Improvements and Results

Hyperparameter tuning on window size



Modalities	Window Past	Window future	F1 Score (%)
atv	1	1	81.72
atv	2	2	83.21
atv	4	4	**84.08**
atv	5	5	83.19
atv	6	6	82.49
atv	7	7	82.28
atv	9	9	82.77
atv	10	10	**84.50**
atv	11	11	83.93
atv	15	15	83.78

Figure 8. Changing window size

The window size can be treated as a hyperparameter that could be adjusted while training our architecture. Moreover, the freedom of setting the window size makes our architecture more flexible in terms of usage. A larger window size would result in better performance for cases where the inter and intra speaker dependencies are maintained for longer sequences. In contrast, setting a lower window size would be better in a use case where the topic frequently changes in dialogues and speakers are less affected by another speaker.

7. Code Repository

https://github.com/salikadave/CSE597CourseProject

References

1) "IEMOCAP- Home." Sail.usc.edu, sail.usc.edu/iemocap/index.html.
2) Majumder, N., et al. "Multimodal Sentiment Analysis Using Hierarchical Fusion with Context Modeling." Knowledge-Based Systems, vol. 161, Dec. 2018, pp. 124–133, https://doi.org/10.1016/j.knosys.2018.07.041.
3) Scotti, Vincenzo, et al. "Combining Deep and Unsupervised Features for Multilingual Speech Emotion Recognition." Lecture Notes in Computer Science, 1 Jan. 2021, pp. 114–128, https://doi.org/10.1007/978-3-030-68790-8_10.
4) Joshi, Abhinav, et al. "COGMEN: COntextualized GNN Based Multimodal Emotion RecognitioN." ArXiv:2205.02455 [Cs], 5 May 2022, arxiv.org/abs/2205.02455.
5) Lab, Exploration. "Exploration-Lab/COGMEN." GitHub, 11 Dec. 2023, github.com/Exploration-Lab/COGMEN.
6) "Papers with Code - IEMOCAP Benchmark (Multimodal Emotion Recognition)." Paperswithcode.com, paperswithcode.com/sota/multimodal-emotion-recognition-on-iemocap. Accessed 12 Dec. 2023.
7) "Torch_geometric.nn — Pytorch_geometric Documentation." Pytorch-Geometric.readthedocs.io, pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.RGCNConv. Accessed 12 Dec. 2023.
8) "Homepage." Comet, comet.ml/. Accessed 12 Dec. 2023.
9) "SentenceTransformers Documentation — Sentence-Transformers Documentation." Www.sbert.net, www.sbert.net/.