CS626 Project - Multimodal Sentiment Analysis

Neel Aryan Gupta, Mohammad Ali Rehan, Shreya Pathak

180050067, 180050061, 180050100

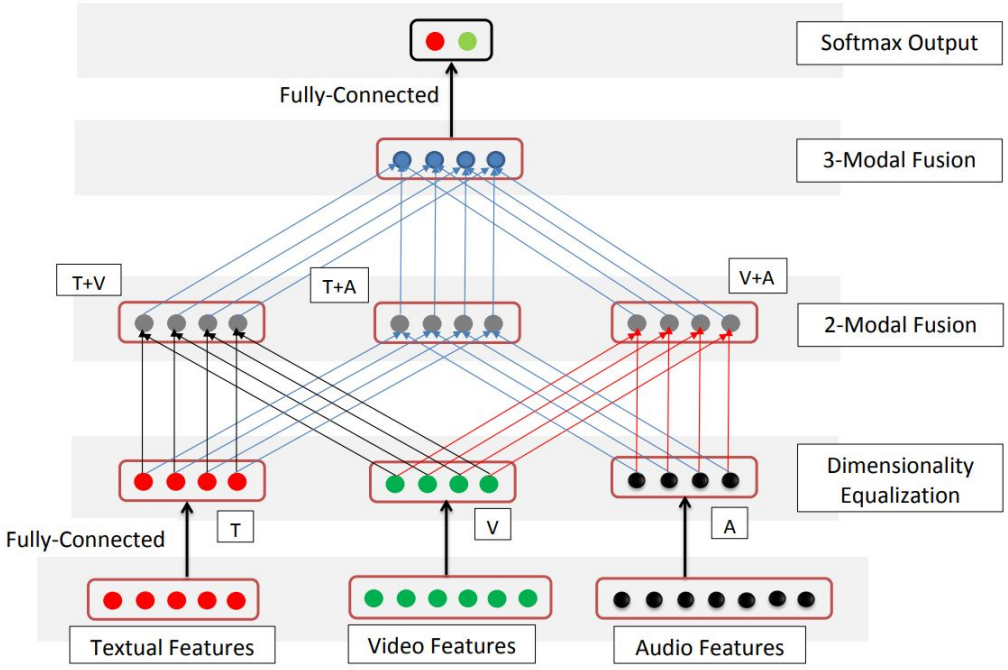
1 Introduction

Multimodal sentiment analysis is the task of using textual visual and audio cues to carry out sentiment analysis on the utternaces of the speaker. Because of discreapncy phenomenon displayed by humans such as sarcasm, the textual cues are not always sufficient to figure out the true sentiment. That is why the process of using other cues become important. Humans also detect sarcasm using voice tone modulation or facial expressions, which we try to use and analyse in this project.
The ideas presented here are mainly taken from the paper here by *N. Majumder et al.* The paper presents an ingenious and interesting model for step wise hierarchical fusion of the multi-modal data for the task at hand.
The collection of data with large companies in the form of spoken user video makes the required data easily available. But till now most architectures have resorted to simple concatenation of data. This is not very effective since the modalities of the data differ a lot.

2 Hierarchical Fusion Networks

2.1 Overview of the Method

The model consists of a unimodal feature extractor. This, as the name suggests take the cues of one modality(either audio, video or text) and convert it into feature vectors that can be used for the next layer. The details are discussed in the following section.
This is followed by a bimodal fusion layer that fuses separately video-audio, audio-text and video-text. This helps the model to find alignment of utterances between two modalities and gives us 3 separate outputs. Finally we use a trifusion layer to combine the three together bimodal cues into one. This can be followed by a fully connected layer and softmax to bring about the necessary classification. A simplified schematic is depicted below



2.2 Unimodal Feature Extractors

Given the raw transcript of the videos, 300-D word2vec is used to obtain word embeddings in most cases. For audio and speech processing we employ covarep tools to extract 74D feature vectors. Citing their home page, Covarep is an open-source repository of advanced speech processing algorithms and is stored as a GitHub project (`https://github.com/covarep/covarep`) where researchers in speech processing can store original implementations of published algorithms. For video, facet features were used.

3 Contextual Relationships between features

Features of each modality are dependent on each other. The tone or meaning of the next utterance is highly continegent on the current word. The pragmatics of a word also depends largely upon the context upon which the word is spoken. Hence we first try to convert the individualistic representations into a form that takes this into account. For this we exploit a GRU (or 3 different GRU's, one for each modality). The equations governing it are shown below:

$$
\begin{aligned}
z_m &= \sigma(f_{mt}U^{mz} + s_{m(t-1)}W^{mz}),\\
r_m &= \sigma(f_{mt}U^{mr} + s_{m(t-1)}W^{mr}),\\
h_{mt} &= \tanh(f_{mt}U^{mh} + (s_{m(t-1)} * r_m)W^{mh}),\\
F_{mt} &= \tanh(h_{mt}U^{mx} + u^{mx}),\\
s_{mt} &= (1 - z_m) * F_{mt} + z_m * s_{m(t-1)},
\end{aligned}
$$

Here s is the information that flows from one time step to the other. z decides the fraction of the contribution made to this information by older information and the rest by the current input received. r can be thought of as *remembrance* and tells that to give the current output, how much weight should be given to information received from the past. This helps in giving h. Finally a dense layer and bias(u) gives us the contextualised output, F. Note that the m stands for modalities and indicated the 3 different GRUs mentioned above.

3.1 Multimodal Fusion Layers

3.1.1 Bimodal layers

We start off by reducing all the feature vector from above (F) to the same size, let's call them g_m ($m \in A, V, T$). At this point the abstract features are powerful enough to carry data like anger of a speaker. We initiate hierarchical fusion by combining pair of modalities in $3C2$ ways. This is done by concatenation of features and then a dense layer to get bimodal fused vectors as below.

$$
\begin{aligned}
i_{lt}^{VA} &= \tanh(w_l^{VA}.[c_{lt}^{V}, c_{lt}^{A}]^\intercal + b_l^{VA}),\\
i_{lt}^{AT} &= \tanh(w_l^{AT}.[c_{lt}^{A}, c_{lt}^{T}]^\intercal + b_l^{AT}),\\
i_{lt}^{VT} &= \tanh(w_l^{VT}.[c_{lt}^{V}, c_{lt}^{T}]^\intercal + b_l^{VT}),
\end{aligned}
$$

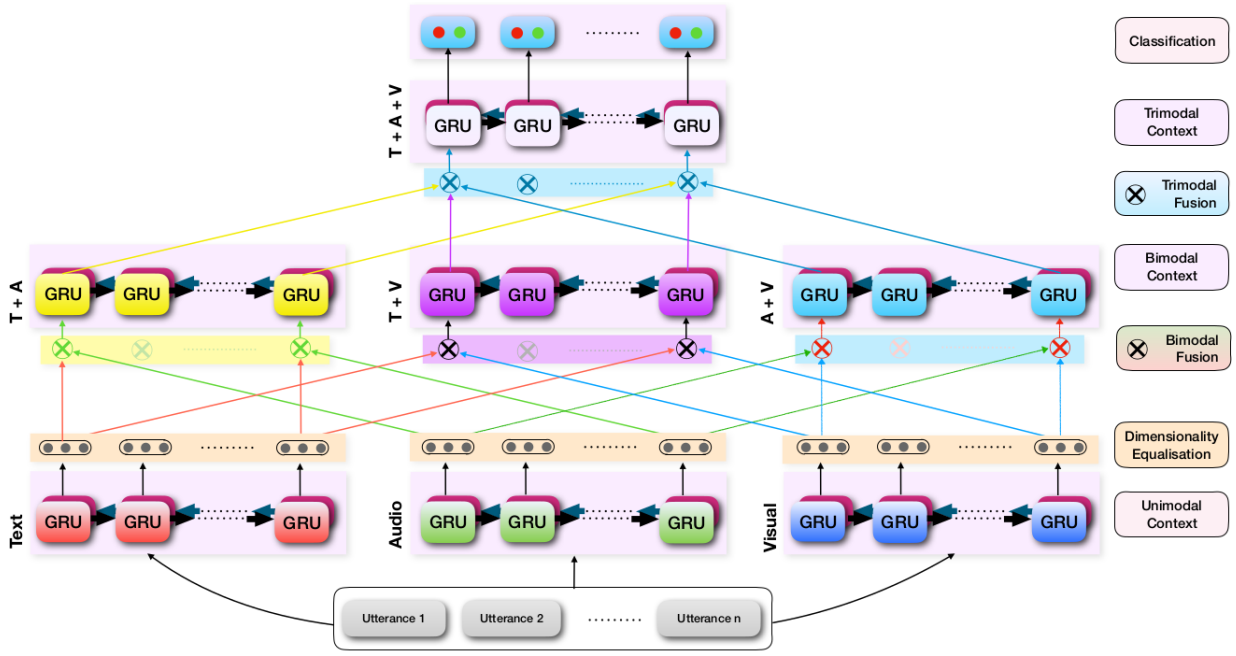
wherein c_{lt}^{m} is a scalar denoting the value in the l^{th} dimension at the t^th time step for the m^th modality.We assemble these values (i_{lt}^{m}) into a vector to get the new bimodal feature vectors $f_t^{mm'}$ where both m, m' are two different modalities. They are finally passed through a GRU for further contextualization. Thus we finally obtain $F_{mm'}$ which is a 2D tensor having the bimodal feature vectors of utterances at all time steps.

3.1.2 Trimodal layers

In the last step of hierarchical fusion we combine all the modalities along their last dimension and then apply a dense layer as shown:

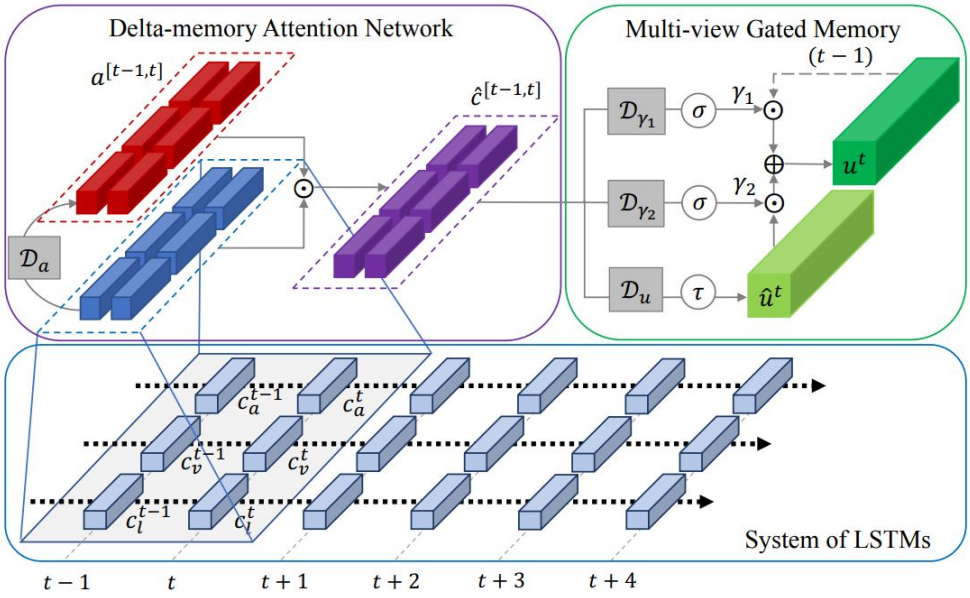
$$z_{lt} = \tanh(w_l^{AVT}.[I_{lt}^{VA}, I_{lt}^{AT}, I_{lt}^{VT}]^\intercal + b_l^{AVT})$$

The z_{lt} is a scalar. We assemble all the $z's$ at a fixed time step (varying the l) into a feature vector f_t which corresponds to the utterance at time step t. Lastly, we apply a final GRU on them to obtain our trimodal fused contextualized feature vectors for the input. We use this for our classification purposes. The diagram below shows the complete schematic of the above procedure.



4 Memory Fusion Networks

MFNs are useful when one needs to predict a single label of sentiment for an entire sequence of utterances. The above model works well when we have a sentiment for every utterance but does not give good result when the task is a sequence to label task. For the latter, one can make use of the MFN architecture to achieve good results. The architecture is described below:



4.1 System of LSTMs

We start by applying an LSTM on all the models separately, so 3 LSTMs in this manner. We are interested in the memory of the LSTMs. This helps to model the fact that the features of an utterance at a later time might depend on the utterance

of the same modality at an earlier time step. This concept forms the basis of all sequence processing models and so is a logical step to begin with. Below are shown equations of LSTM:

$$
\begin{aligned}
i_n^t &= \sigma(W_n^i\ x_n^t + U_n^i\ h_n^{t-1} + b_n^i) \\
f_n^t &= \sigma(W_n^f\ x_n^t + U_n^f\ h_n^{t-1} + b_n^f) \\
o_n^t &= \sigma(W_n^o\ x_n^t + U_n^o\ h_n^{t-1} + b_n^o) \\
m_n^t &= W_n^m\ x_n^t + U_n^m\ h_n^{t-1} + b_n^m \\
c_n^t &= f_n^t \odot c_n^{t-1} + i_n^t \odot m_n^t \\
h_n^t &= o_n^t \odot tanh(c_n^t)
\end{aligned}
$$

i_n, f_n, o_n are the input,forget and output gates. c_n is the memory of LSTM, h_n is the output. We combine the previous output and current input to form current state(m_n). m_n is also called proposed memory update. New memory has some fraction of older memory and rest is current update. Output is decided by suppressing some memory values.

4.2 Delta-memory Attention network

Its purpose is to model the relationships between the different modalities of an utterance. To account for the fact that relationship between current modalities may be governed by previous states, we jointly use the memory at the current and previous time stage. This allows us to include how a particular element in the memory vector changed at the current time step. That's why its called a delta network. Had we not included the previous time step, the layer would be biased to give more attention to any large values in the current time step. However with the previous time step included, it can learn to give importance to changes in memory states, rather than a large unchanged value propagating through the LSTM. Mathematically,

$$\hat{c}^{[t-1,t]} = D_a(c^{[t-1,t]}) * c^{[t-1,t]}$$

wherein $\hat{c}^{[t-1,t]}$ are attended LSTM memories. D_a is a dense attention layer with softmax activation of output size same as input size, i.e. twice of LSTM memory size) and $c^{[t-1,t]}$ is concatenation of the the two LSTM memories. $*$ denotes elementwise product.

4.3 Multi-view Gated Memory

Now that we have cross view interactions, we would like to link them over time. This could be done by simply passing all of them through a GRU/LSTM however we use something a notch above. The Multi-view Gated Memory is controlled using set of two gates. γ_1, γ_2 are called the retain and update gates respectively. At each time-step t, γ_1 assigns how much of the current state of the Multi-view Gated Memory to remember and γ_2 assigns how much of the Multi-view Gated Memory to update based on the update proposal $\hat{u}^t$. $\hat{u}^t$ is obtained by applying a dense layer to $\hat{c}^{[t,t-1]}$. Hence,

$$u^t = \gamma_1^t * u^{t-1} + \gamma_2^t * tanh(\hat{u}^t)$$

$tanh$ is used to control the magnitudes of u.$*$ is element wise multiplication. Multi-view Gated Memory is an advancement over LSTM, as here the retain and update gates are themselves obtained by applying a neural network on $\hat{c}^t$, whereas in LSTM we use only an affine transformation.

4.4 Output

The consolidated output is the concatenation of u^T and h^T of all modalities(output of LSTM). We can pass this through a hidden layer and then an output layer to get the sentiment probabilities.

5 Dataset Details

5.1 Preprocessing

We employ a technique called alignment. To work with multimodal time series that contains multiple views of data with different frequencies, we have to first align them to a pivot modality. The data in the raw form consists of utterances sampled at different frequencies in different modalities. What is done is that we take one modality and use its time steps to split the other modalities samples into bins. Each bin may have more than one samples, and hence we apply a collating function (like average) that combines it into a single feature corresponding to that time step of the chosen modality. In word aligned modality, we use the word features for this purpose. This gives rise to a sequence to sequence data-set that is suitable for hfusion type models. For the label aligned setting, wherein features of many utterances fall together as a group under a

single sentiment, the MFN type sequence to label models give better result. Before 2018, only label aligned data sets were popular, such as those used by MFNs. Later utterance aligned also began to be used.

5.2 Dataset sizes

5.2.1 Word Aligned

IEMOCAP stands for Interactive Emotional Dyadic Motion Capture. It was developed by SAIL at USC. It has 4 sentiment labels corresponding to anger, happiness, sadness, neutrality; as given by multiple annotators.

IEMOCAP	Size	Seq. Length	Text	Audio	Video	#Labels
Train	120	110	100	100	100	4
Test	30	110	100	100	100	4

MOSEI stands for Multimodal Opinion Sentiment and Emotion Intensity. It was released by CMU and has information about both the sentiment(happy/angry/sad/digusted/surprised/ fearful) along with the intensity of this emotion. We divide the sentiment into 3 categories: Positive, Neutral and Negative and use this.

MOSEI	Size	Seq Length	Text	Audio	Video	#Labels
Train	2150	98	300	74	35	2
Test	100	98	300	74	35	2

5.2.2 Label Aligned

Label aligned IEMOCAP version has a sequence length of 20 and 4 classes.

IEMOCAP	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	7878	21	300	74	35	4
Test	1970	21	300	74	35	4

MOSI stands for Multimodal Corpus of Sentiment Intensity. It was released by CMU. Each opinion video is annotated with sentiment in the range [-3,3]. There exist two version having sequence length of 20 and 50 utterances. We use the 20 version and divide the sentiments into 2 classes, positive and negative.

MOSI	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	1682	20/50	300	5	20	2
Test	421	20/50	300	5	20	2

We use label aligned MOSEI (Multimodal Opinion Sentiment and Emotion Intensity) of 20 sequence length, made available by CMU.

MOSEI	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	15913	20	300	74	35	3
Test	3979	20	300	74	35	3

MMMO (Multi-Modal Movie Opinion) is relatively smaller dataset made from spoken movie review videos.

MMMO	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	248	21	300	74	35	2
Test	63	21	300	74	35	2

YouTube dataset has sentiments associated with a subset of YouTube videos.

Youtube	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	215	21	300	74	35	3
Test	54	21	300	74	35	3

MOUD stands for Multimodal Opinion Utterance Dataset. This is a collection of video reviews, created by university of Michigan

MOUD	#Annotations	Seq. Length	Text	Audio	Video	#Labels
Train	308	21	300	74	35	3
Test	78	21	300	74	35	3

6 Observations & Analysis

6.1 Hierarchical fusion model

As mentioned above, we trained and tested our models on two datasets. To observe the effects of addition of modalities we present the statistics of unimodal (only text), bimodal (only text and audio) and trimodal (all three) models

6.1.1 IEMOCAP

The experiemental accuracy values for unimodal version utilising only textual features is shown below:

GRU Memory size→	500	400	600
Unimodal	73.0	72.5	66.9

For the bimodal case which uses text and audio
Note here the parameters refer to the projection dimension of the 2 Dense layers and the GRU Memory size

Parameters→	(250, 250, 350)	(200, 200, 350)	(200, 100, 350)
Bimodal	78.0	77.9	76.7

We run hyper-parameter search over the memory size of the final GRU. Note that in a broad of parameters the parameters didn't bring about any significant change in accuracy

Parameters→	400	450	500	600
Trimodal	78.5	79.8	78.89	79.1

The best achieved accuracies for each of the modalities is shown below.

Modality / Metric	Unimodal	Bimodal	Trimodal
Accuracy	0.73	0.78	0.80
Macro F1	0.72	0.77	0.80

For the IEMOCAP dataset, the best results are shown below side by side for each of the models to aid comparison

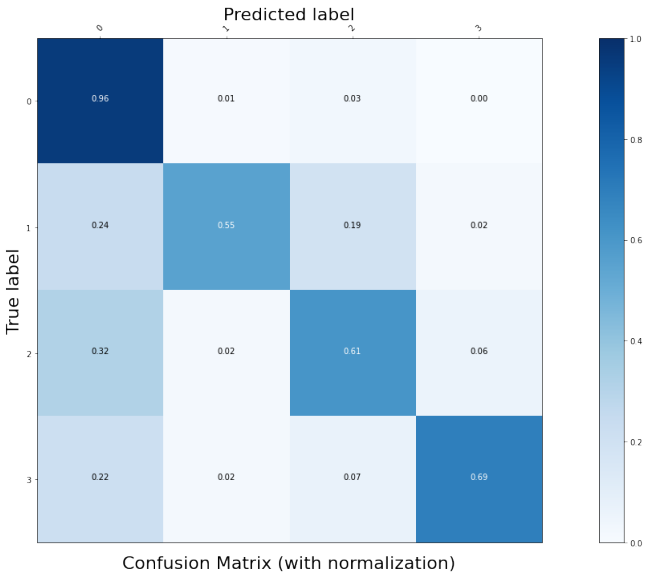


Figure 1: Unimodal

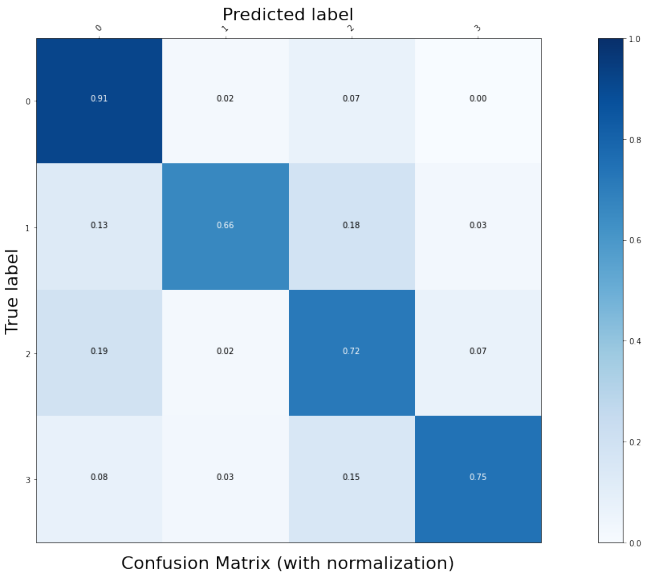


Figure 2: Bimodal

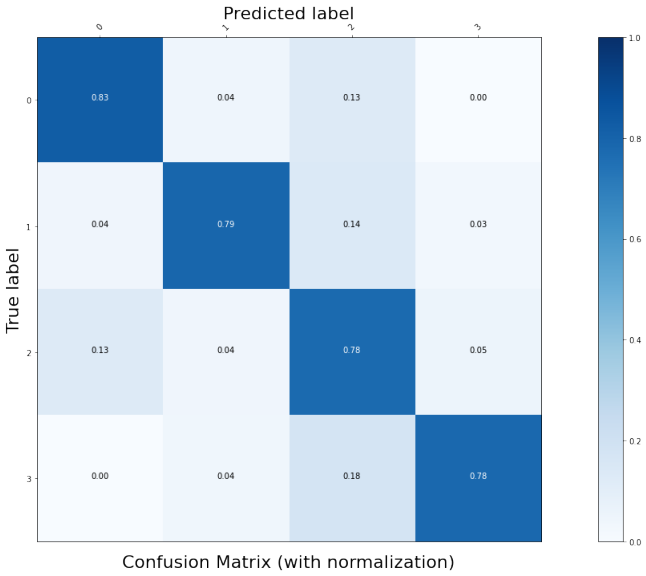


Figure 3: Trimodal Confusion Matrix

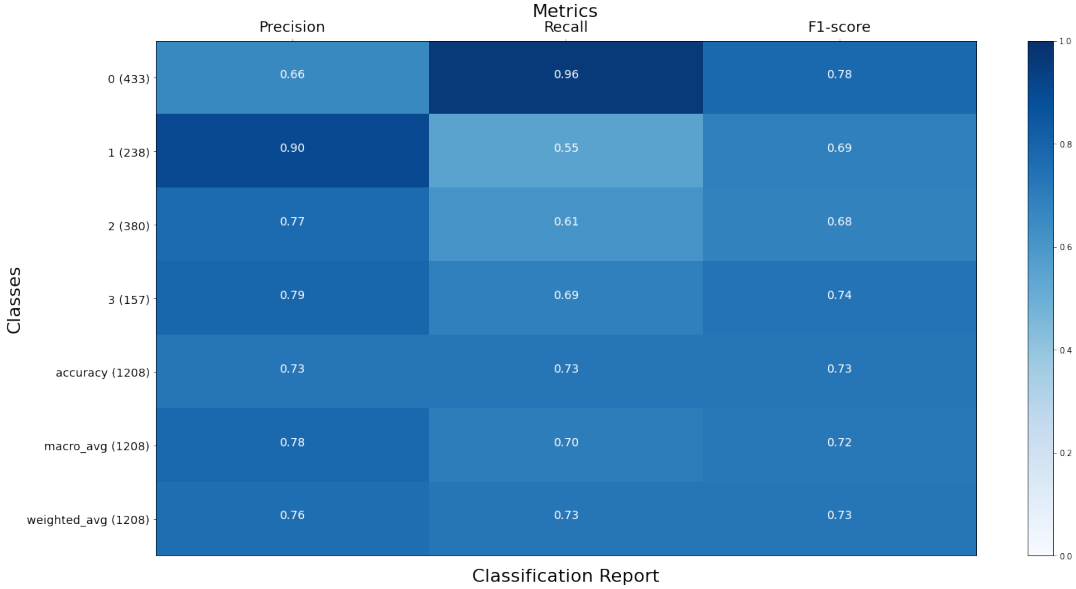


Figure 4: Unimodal

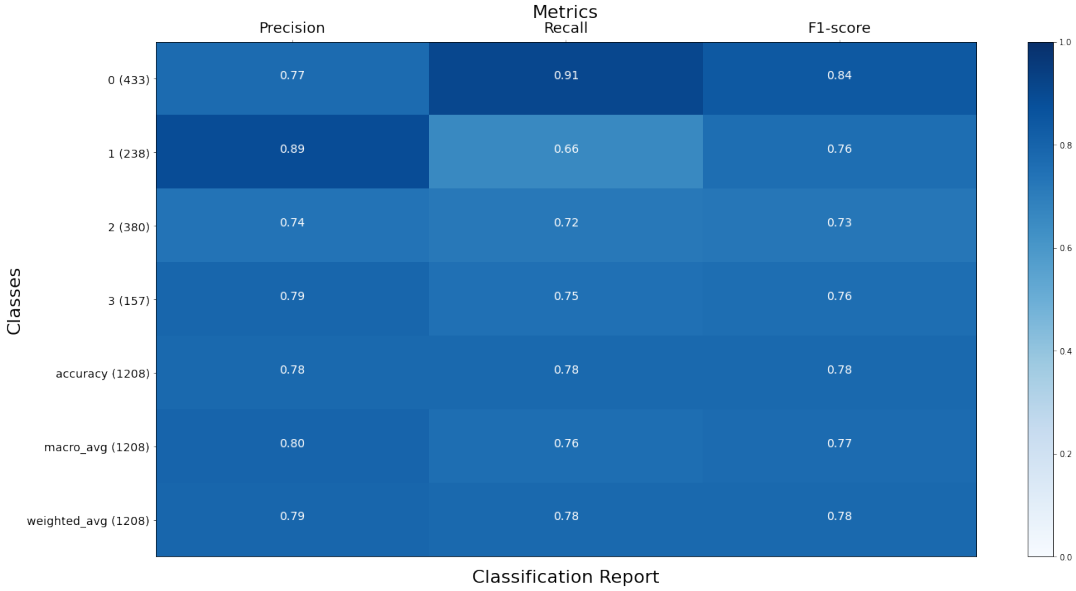


Figure 5: Bimodal

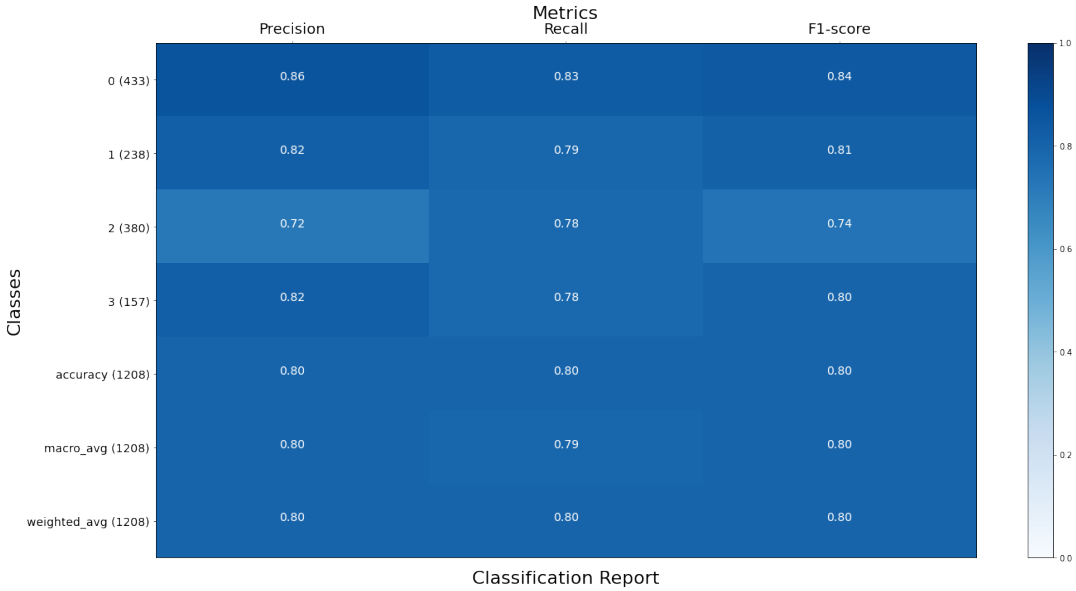


Figure 6: Trimodal
Classification Report

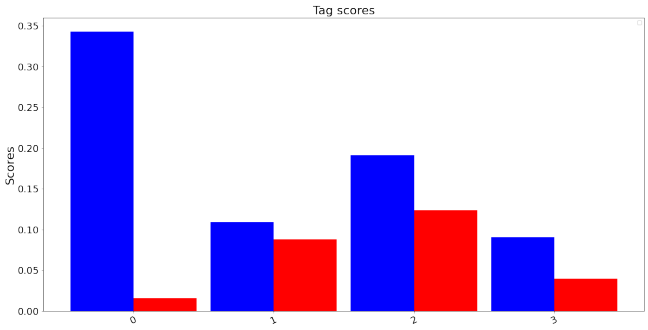


Figure 7: Unimodal

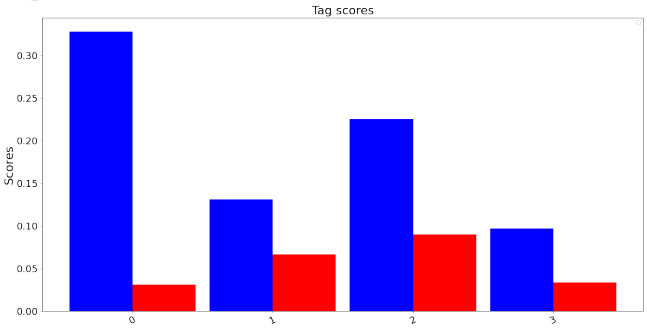


Figure 8: Bimodal

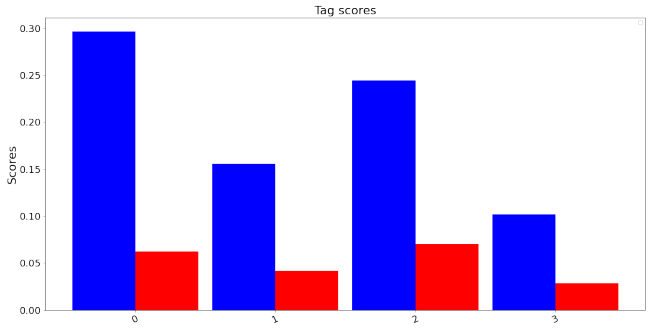


Figure 9: Trimodal
Tag Scores

There is considerable increase in accuracy by the addition of simple statistical audio features. We observe diminishing returns when video is added showing audio and text is sufficient for sentiment analysis. The role that audio plays is that we often express pragmatics (such as sarcasm) by tone and pitch modulation. The F1 score can be observed to increase also. Specifically, the recall of class 0 decreases and precision increases. This shows that the model had a bias towards predicting 0 when enough modalities were not available, and this considerably reduces by adding auditory and visual cues.

6.1.2 MOSEI

The experiemental accuracy values for unimodal version utilising only textual features is shown below:

GRU Memory size→	300	250	350
Unimodal	70.9	70.1	70.3

For the bimodal case which uses text and audio
Note here the parameters refer to the projection dimension of the 2 Dense layers and the GRU Memory size

Parameters→	(200, 100, 300)	(250, 100, 300)	(200, 100, 350)
Bimodal	75.0	72.7	73.3

In the trimodal case we searched over the memory size of last GRU For the MOSEI dataset, the results are shown below

Parameters→	400	450	500	600
Trimodal	78.7	78.9	80.1	79.7

The best achieved accuracies for each of the modalities is shown below.

Modality / Metric	Unimodal	Bimodal	Trimodal
Accuracy	0.71	0.75	0.80
Macro F1	0.63	0.72	0.77

For the MOSEI dataset, the best results are shown below side by side for each of the models to aid comparison

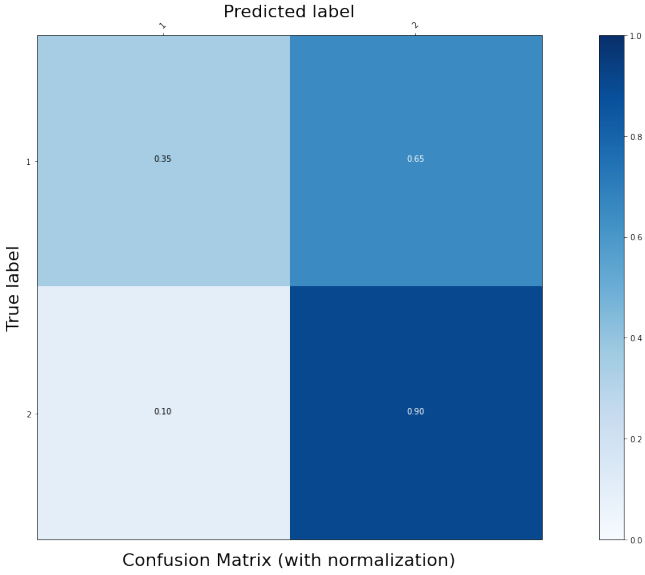


Figure 10: Unimodal

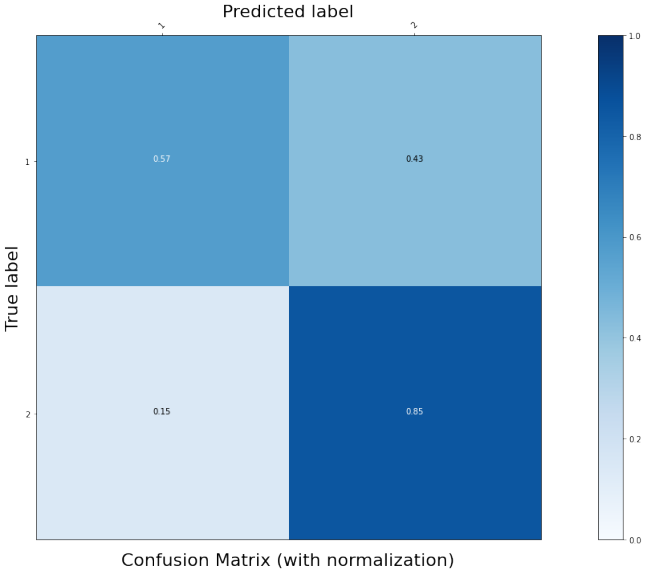


Figure 11: Bimodal

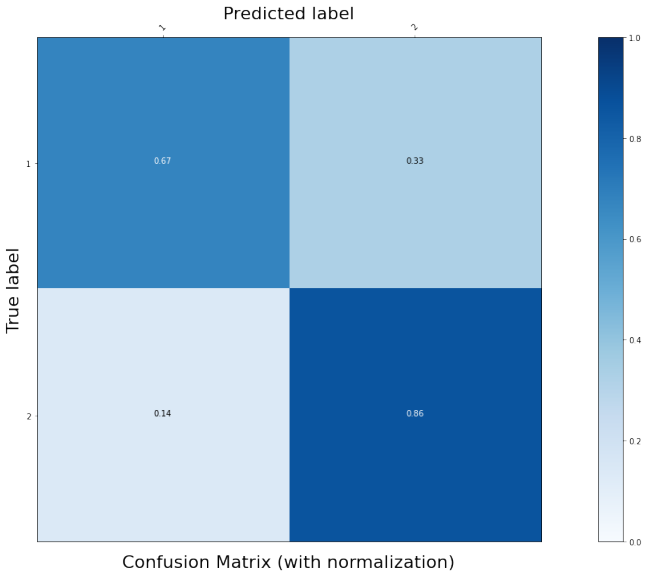


Figure 12: Trimodal Confusion Matrix

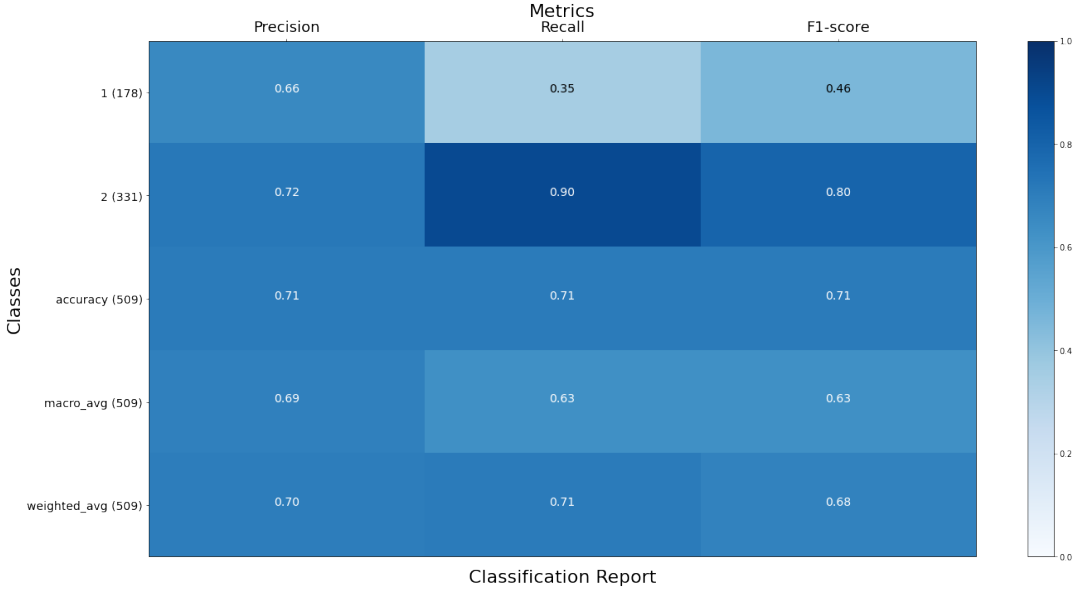


Figure 13: Unimodal

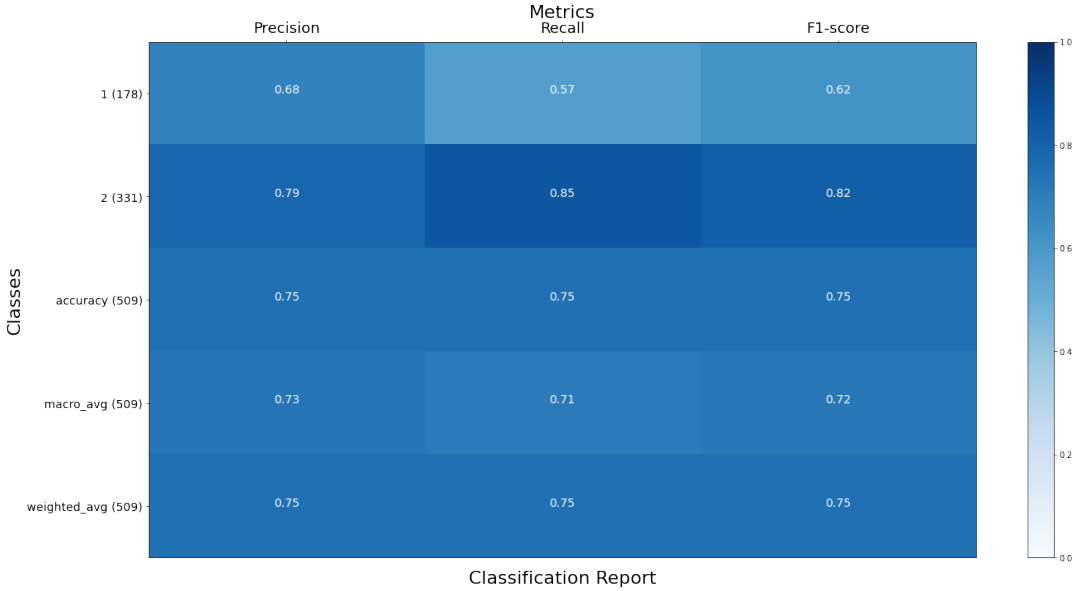


Figure 14: Bimodal

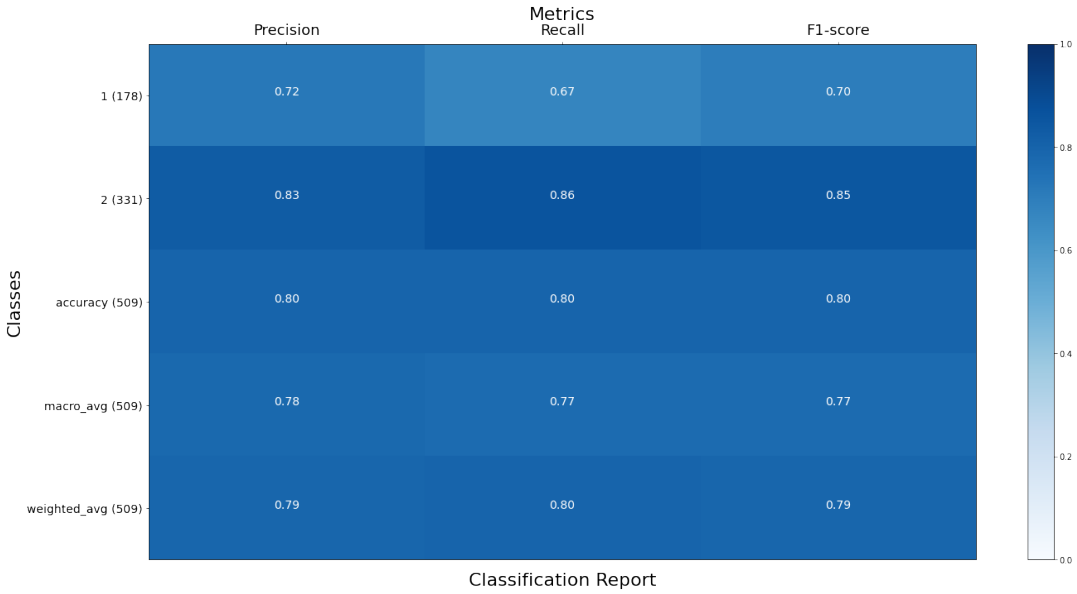


Figure 15: Trimodal
Classification Report

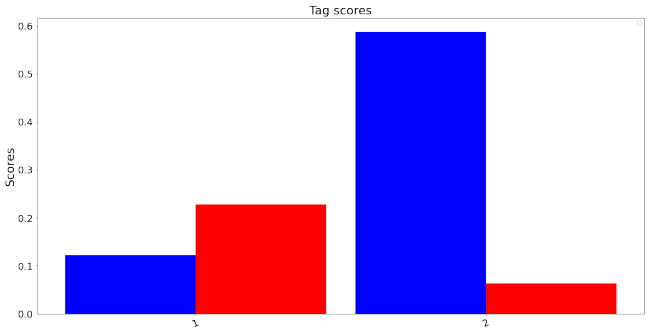


Figure 16: Unimodal

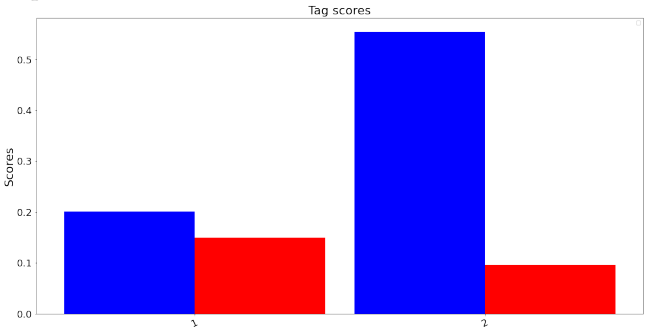


Figure 17: Bimodal

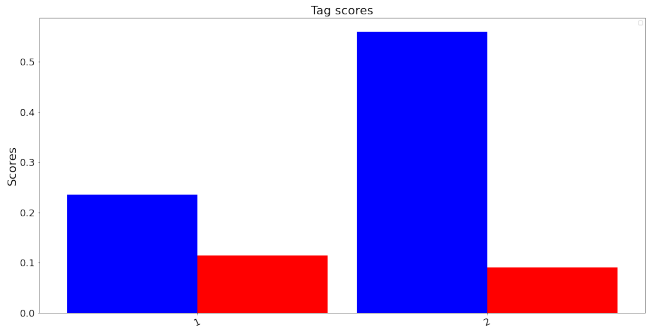


Figure 18: Trimodal
Tag Scores

In the MOSEI dataset that we used the 0 label was orders of magnitude more than the other 2 labels. Because of this, the model was mostly learning to predict class mostly as 0 and giving unrealistic accuracy scores. On masking 0 classes, we are able to get more meaningful results. Here also it was observed that number of examples for class 1 was about half than that of class 2. Thus when using only text one observes a bias towards predicting 2. This can be seen in the almost 2 fold increase in F score upon adding audio features. Both per class accuracy scores improve and recall of class 2 decreases as the model now predicts more smartly, mistaking some extra class 2 examples but overall predicting with increased accuracy.

6.2 Memory Fusion Network

6.2.1 IEMOCAP

We perform hyper-parameter search over relevant memory sizes of the multi view gated network. The following table shows some of the collected accuracy scores over different memory sizes:

Modality \ Memory Size	128	256	512
T	0.53	0.54	0.53
T+A	0.54	0.56	0.56
T+A+V	0.55	0.57	0.57

This table shows the macro-averaged F1 scores corresponding to the above table:

Modality \ Memory Size	128	256	512
T	0.37	0.33	0.43
T+A	0.42	0.47	0.50
T+A+V	0.48	0.52	0.53

The best value comes out to be 512, beyond that overfitting occurs quickly. For 512, the accuracy and macro-averaged F1 scores of all possible combinations of modalities are shown below:

Metric \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
Accuracy	0.53	0.53	0.50	0.56	0.54	0.53	0.57
Macro F1	0.43	0.32	0.23	0.50	0.45	0.36	0.53

For the IEMOCAP dataset, the results are shown below

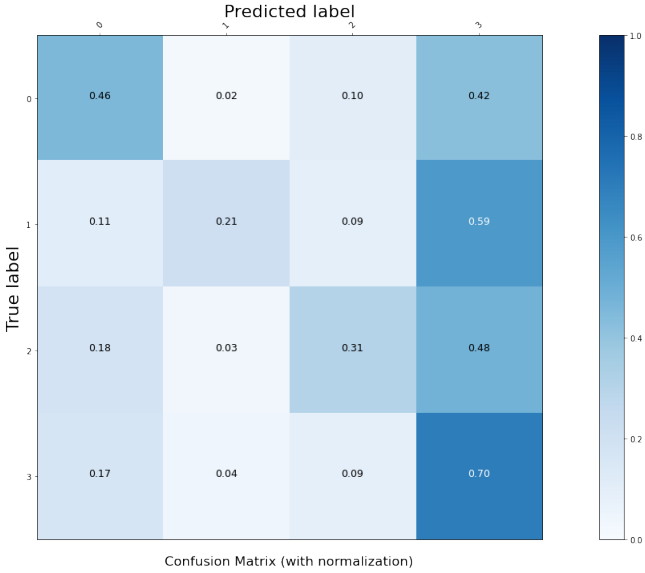


Figure 19: Unimodal

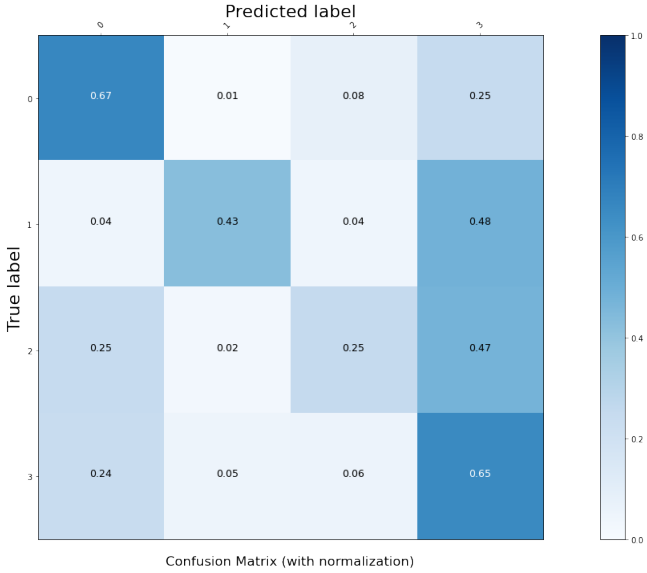


Figure 20: Bimodal

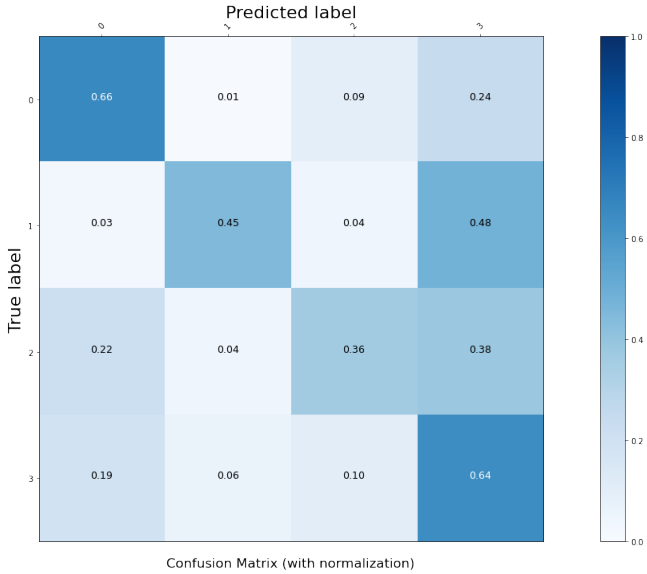


Figure 21: Trimodal Confusion Matrix

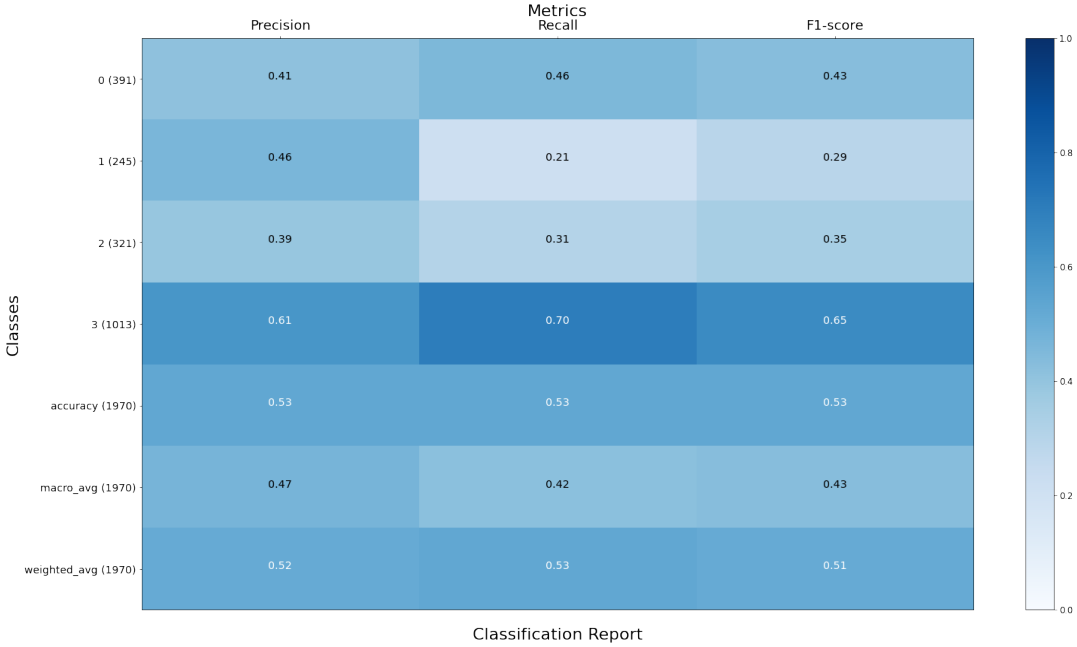


Figure 22: Unimodal

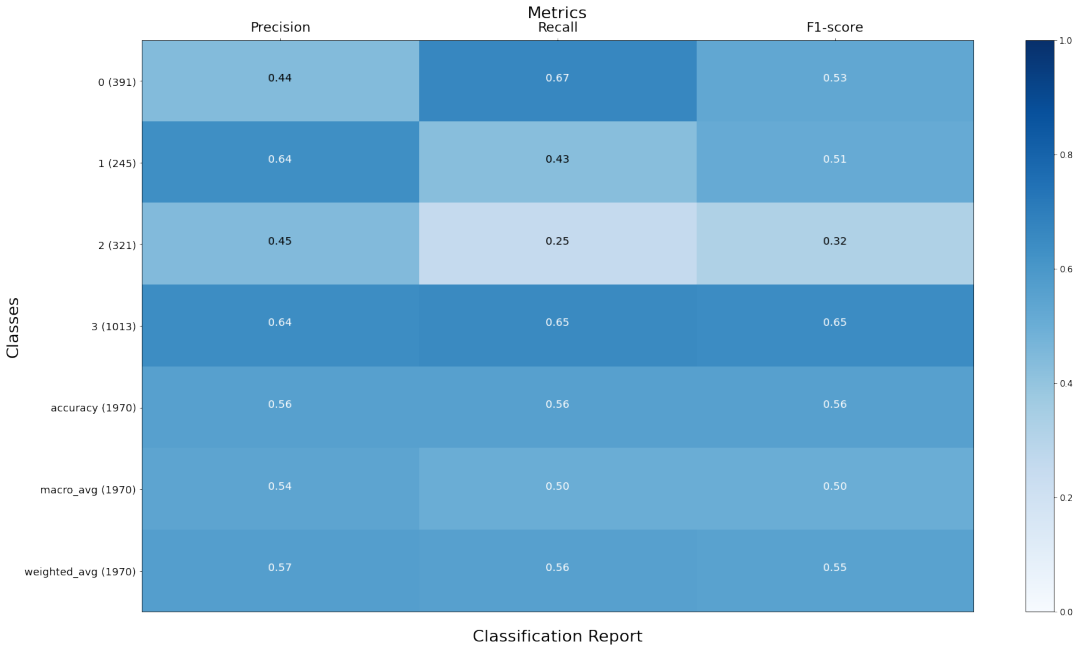


Figure 23: Bimodal

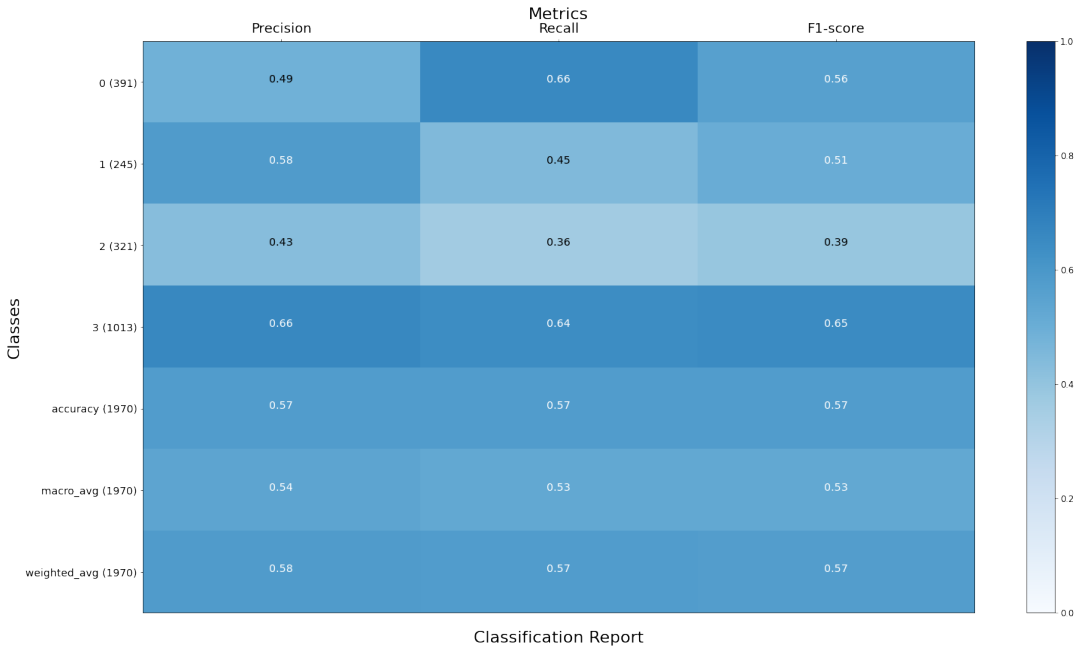


Figure 24: Trimodal
Classification Report

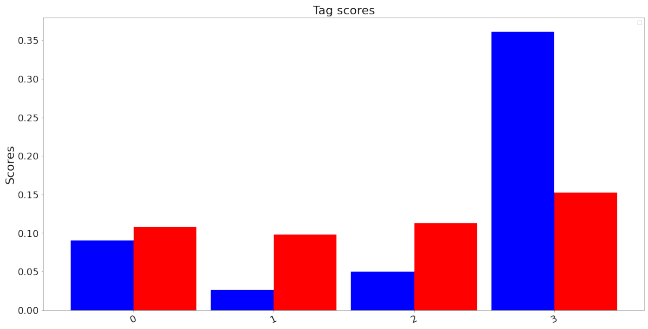


Figure 25: Unimodal

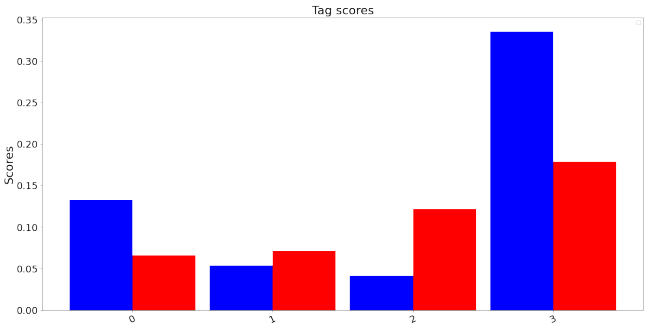


Figure 26: Bimodal

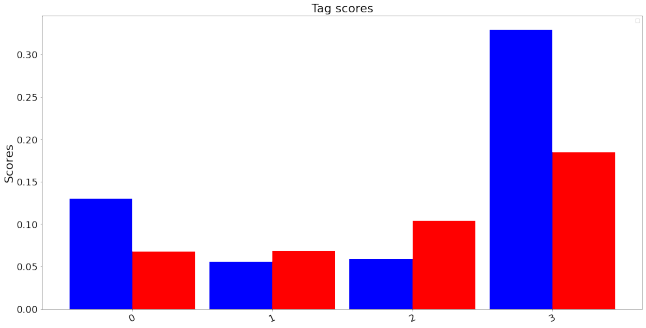


Figure 27: Trimodal
Tag Scores

6.2.2 MOSEI

The optimal memory size comes out around 512 as in the case of IEMOCAP. We show the accuracy and macro-averaged F1 scores of various modalities below:

Metric \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
Accuracy	0.60	0.58	0.57	0.60	0.62	0.58	0.62
Macro F1	0.51	0.41	0.44	0.53	0.53	0.45	0.54

For the MOSEI dataset, the results are shown below

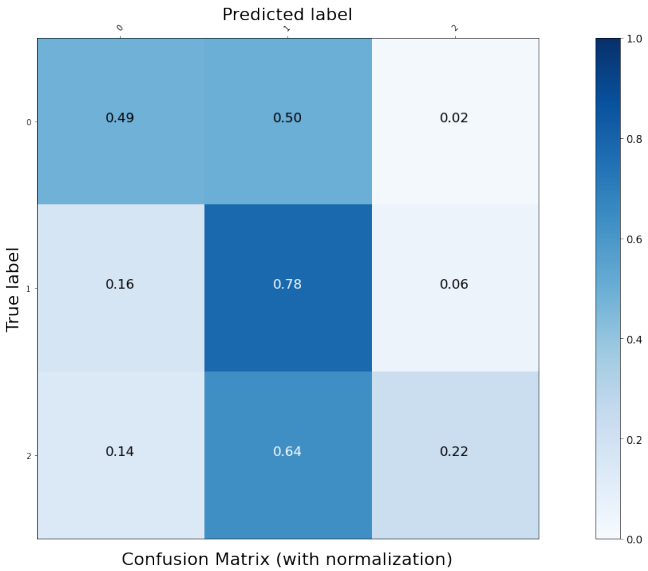


Figure 28: Unimodal

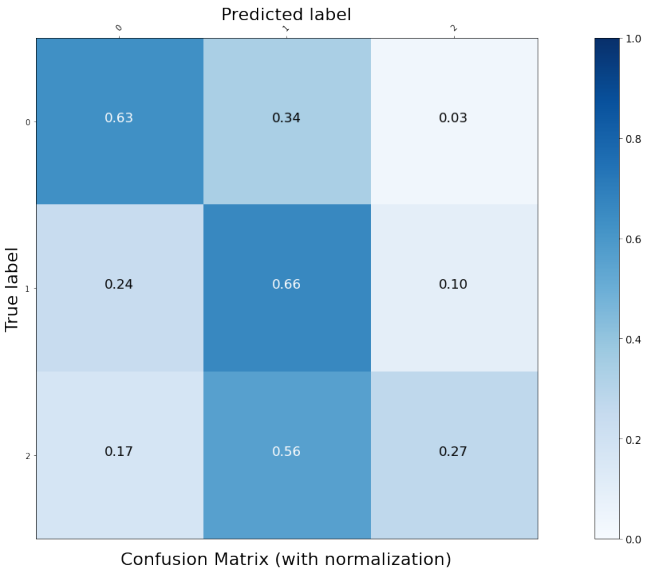


Figure 29: Bimodal

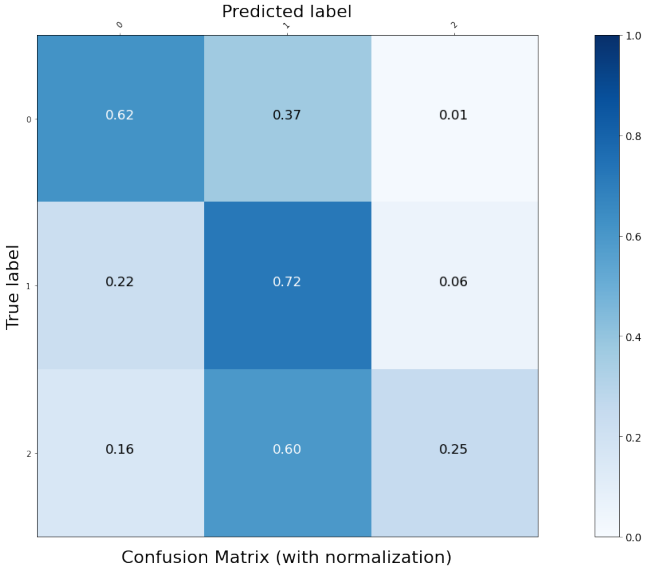


Figure 30: Trimodal Confusion Matrix

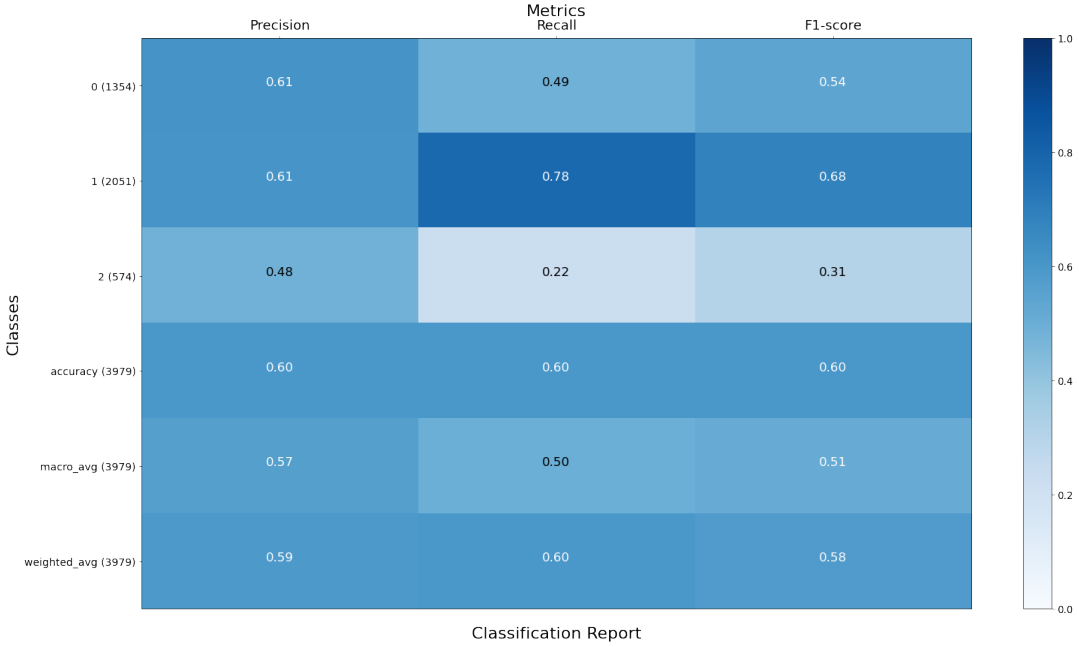


Figure 31: Unimodal

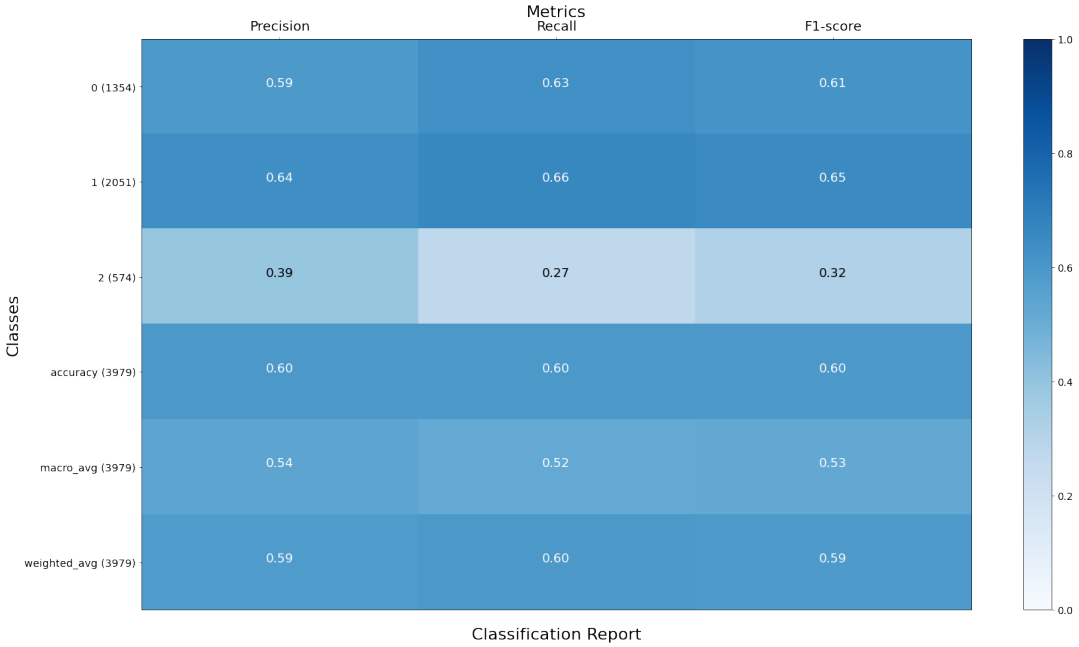


Figure 32: Bimodal

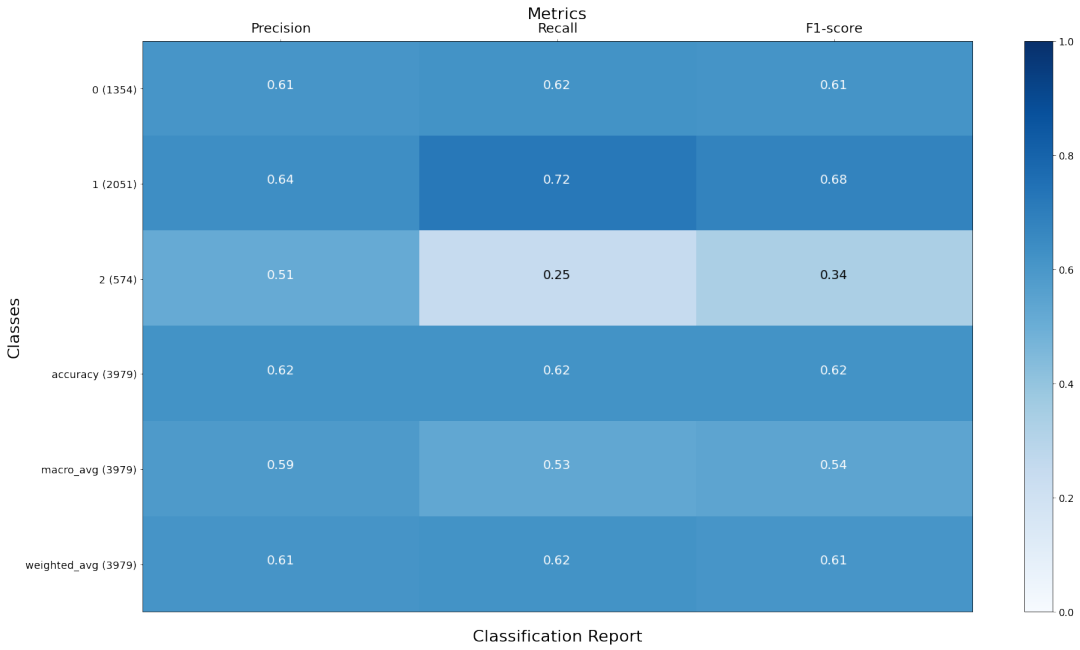


Figure 33: Trimodal
Classification Report

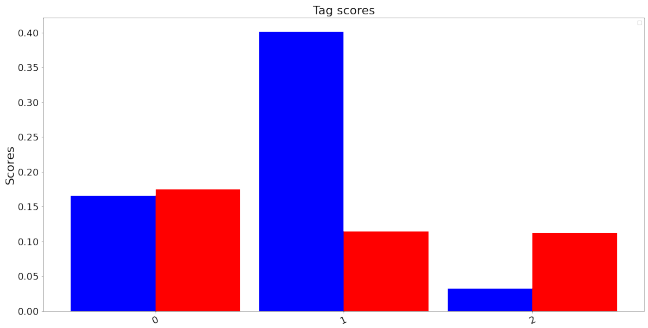


Figure 34: Unimodal

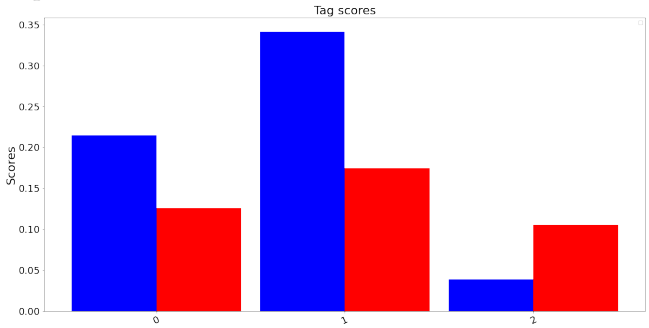


Figure 35: Bimodal

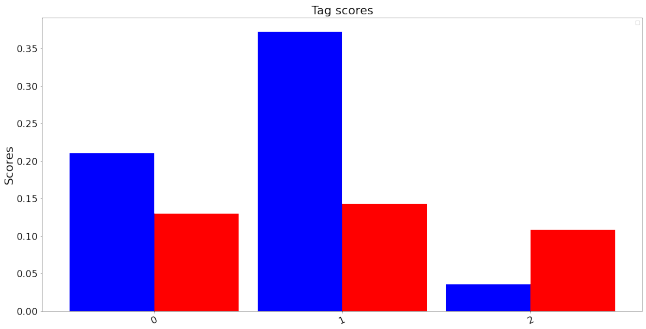


Figure 36: Trimodal
Tag Scores

The same phenomenon can be observed here wherein classes with a smaller number of test examples are continuously improving by adding modalities. This is seen as the increase in F1 score of class 2. The textual modality is the dominant factor here and we can see how its addition to audio and video separately increases their accuracy. One reason for this could be that MOSEI has 300D word2vec embeddings which have far more information about the words than what could be derived from the data-set about other modalities.

6.2.3 MOSI

We train our model on 2 different versions of the MOSI (Multimodal Corpus of Sentiment Intensity) dataset, with each data sample having 20 and 50 sequence length.
Here are the accuracy and macro-averaged F1 scores for 20 sequence length version :

Metric \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
Accuracy	0.71	0.47	0.51	0.73	0.72	0.51	0.74
Macro F1	0.71	0.39	0.50	0.73	0.72	0.47	0.74

Here are the accuracy and macro-averaged F1 scores for 50 sequence length version :

Metric \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
Accuracy	0.70	0.49	0.48	0.73	0.73	0.51	0.76
Macro F1	0.70	0.45	0.43	0.73	0.73	0.49	0.75

6.2.4 5-Fold Validation

3 datasets namely MMMO (Multi-Modal Movie Opinion), MOUD (Multimodal Opinion Utterances Dataset) and YouTube datasets as made available by CMU. 5-Fold cross validation results have been presented on these datasets, as these are quite small datasets. We present both the accuracy scores and macro-F1 scores which have been averaged over 5 folds.
Here are the accuracy scores :

Dataset \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
MMMO	0.68	0.73	0.65	0.78	0.76	0.74	0.80
MOUD	0.59	0.53	0.56	0.60	0.63	0.59	0.64
YouTube	0.46	0.38	0.40	0.47	0.47	0.42	0.48

Here are the (macro-averaged) F1 scores :

Dataset \ Modality	T	A	V	T+A	T+V	A+V	T+A+V
MMMO	0.64	0.69	0.62	0.74	0.73	0.69	0.75
MOUD	0.49	0.42	0.53	0.55	0.61	0.57	0.63
YouTube	0.44	0.24	0.24	0.44	0.42	0.28	0.44

6.3 Significance Testing

In some cases where the accuracy did not increase very much we do significance testing. We use the χ^2-test as proposed by Pearson. The test says that we start with a table(T) of observed counted with observed classes in the columns, and true classes in the rows. Lets call the entries (O_{ij}). The expected counts(E_{ij}) can for a particular cell can be found by multiplying the column and row sum of this cell in T and dividing by sum of entries in T. The test says that

$$\sum_{ij} \frac{(E_{ij} - O_{ij})^2}{E_{ij}} \sim \chi_c^2$$

wherein $c = (\#cols_T - 1)(\#rows_T - 1)$. The count is based on table T

When the value is larger do we say that the behaviour is highly different from random (independent behaviour) and hence there is some significance of our result. We carry out this procedure for combinations where a significant increase in accuracy is not seen. Below table shows test score for various cases that we chose to study

Datatset	Model	Unimodal	Bimodal	Trimodal
IEMOCAP	Hfusion	1554.53 (73)	1794.94 (78)	1982.28 (80)
IEMOCAP	MFN	396.79 (53)	846.41 (56)	930.64 (57)
MOSEI	MFN	704.62 (60)	797.61 (60)	949.385 (62)

Value in brackets is accuracy in percentage
We can use the above table to compare cases like IEMOCAP Bimodal and Trimodal wherein the change in accuracy is not large but the test statistic report an increase in performance. Same is the case with MOSEI Unimodal and Bimodal wherein the accuracy almost remained same but the χ^2-score improved.

7 Drawbacks

1. The 300D word embedding contain much more information than can be gained by just analysing the video and audio signals. This sometime causes the models to overpower the other modalities and focus more on text than other modalities, even when they might have more information when dealing with pragmatics
2. Small datasets like Youtube and MMO can give rise to results with high variance as the models are quite large. Hence it takes a lot of tuning to ensure that overfitting is not taking place.
3. When dataset is highly skewed, as in the case of word aligned MOSEI, the model learns to predict in a naive way by labelling all examples as 0.
4. In MFN models, the number of examples of a class need to be proportional to other classes, otherwise the model does not learn anything about the underrepresented classes at all.

8 Datasets & References

8.1 Datasets

Most of our datasets can be found at:

- `http://immortal.multicomp.cs.cmu.edu/raw_datasets/processed_data/`
- `http://immortal.multicomp.cs.cmu.edu/cache/`

Useful github repository for preprocessing datatsets:
`https://github.com/A2Zadeh/CMU-MultimodalSDK`

SenticNet repository from where we studied and understood and implemented trimodal fusion code:
`https://github.com/SenticNet/hfusion`

Label aligned data (used to train MFN) has been collected at:
`https://www.kaggle.com/neelaryan/multimodal-sentiment-analysis/`

8.2 References

[1] N. Majumdera, D. Hazarikab, A. Gelbukha, E. Cambriac, S. Poria
Multimodal Sentiment Analysis using Hierarchical Fusion with Context Modeling
`https://arxiv.org/abs/1806.06228`

[2] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, Louis-Philippe Morency
Memory Fusion Network for Multi-view Sequential Learning
`https://arxiv.org/abs/1802.00927`