NATURAL LANGUAGE PROCESSING - CSE556

MultiModal Emotion Analysis - ReadMe

Hamzah Akhtar
2018051

Shikhar Sheoran
2018099

Sarthak Pal
2018412

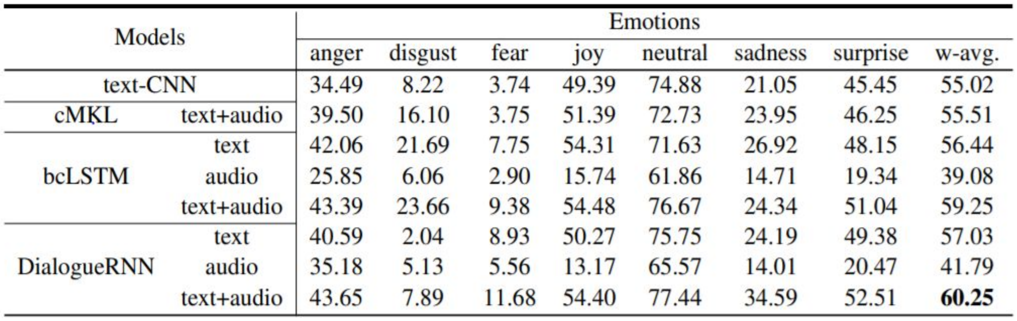
Problem Statement:

Our task is to do emotion analysis on the **MELD: Multimodal EmotionLines Dataset**. MELD has more than **1400 dialogues** and **13000 utterances** from the Friends TV show. In MELD; **Text**, **Audio**, and **Video**; the 3 modalities of each dialogue were used to annotate the utterances with the most appropriate emotion category (**Joy**, **Sadness**, **Fear**, **Surprise**, **Disgust**, **Neutral** and **Anger**) and sentiment category (**positive**, **negative**, **neutral**). We use this as our dataset to form and train **multimodal algorithms** for the task of **emotion classification**. We use two modalities, **text** and **audio**. We report the **F1-Score** and **weighted accuracy** of our model.

Background Work Done related to the problem statement

1. **MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations**

   In this paper, MELD, a multimodal multi-party conversational emotion recognition dataset was developed. We use this, as our base dataset for our project. MELD contains raw videos, audio segments, and transcripts for multimodal processing. Results for the emotion classification on MELD have been shown. The table shows the performance of DialogueRNN, whose multimodal variant achieves the best performance (67.56% F-score) surpassing multimodal bcLSTM (66.68% F-score). Multimodal DialogueRNN also outperforms its unimodal counterparts.



Models		Emotions							
		anger	disgust	fear	joy	neutral	sadness	surprise	w-avg.
text-CNN		34.49	8.22	3.74	49.39	74.88	21.05	45.45	55.02
cMKL	text+audio	39.50	16.10	3.75	51.39	72.73	23.95	46.25	55.51
bcLSTM	text	42.06	21.69	7.75	54.31	71.63	26.92	48.15	56.44
	audio	25.85	6.06	2.90	15.74	61.86	14.71	19.34	39.08
	text+audio	43.39	23.66	9.38	54.48	76.67	24.34	51.04	59.25
DialogueRNN	text	40.59	2.04	8.93	50.27	75.75	24.19	49.38	57.03
	audio	35.18	5.13	5.56	13.17	65.57	14.01	20.47	41.79
	text+audio	43.65	7.89	11.68	54.40	77.44	34.59	52.51	**60.25**

Table 11: Test-set weighted F-score results of DialogueRNN for emotion classification in MELD. Note: *w-avg* denotes weighted-average. text-CNN and cMKL: contextual information were not used.

2. **DialogueRNN: An Attentive RNN for Emotion Detection in Conversations**

   In this paper, an RNN based neural architecture was developed for emotion detection. They have assumed that the emotion of an utterance depends on 3 factors: the speaker, the context given by the previous utterances, and the emotion behind the previous utterances. Two datasets were used for training and testing, namely IEMOCAP and AVEC. DialogueRNN outperforms CMN on both the datasets. For the IEMOCAP dataset, DialogueRNN surpasses CMN with 2.77% accuracy and 3.76% F1-Score, on average. For the AVEC dataset, DialogueRNN yields significantly lower mean absolute error(MAE), and a higher Pearson correlation coefficient.

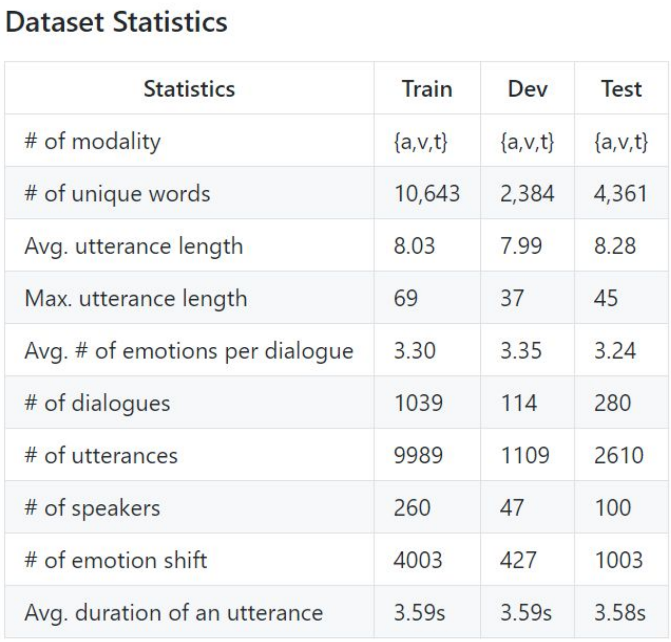
3. **Fusing Audio, Visual and Textual Clues for Sentiment Analysis from Multimodal Content**

   This paper presents a novel method for multimodal sentiment classification, which uses both feature and decision-level. Fusion methods were used to merge effective information extracted from multiple modalities. The dataset used was from Youtube. In the dataset, various videos were collected from Youtube on different topics. Several supervised classifiers, namely Naive Bayes, SVM, ELM, and Neural Networks, were employed on the fused feature vector to obtain each video segment's sentiment. The best accuracy was obtained using the ELM classifier.

Dataset Used:

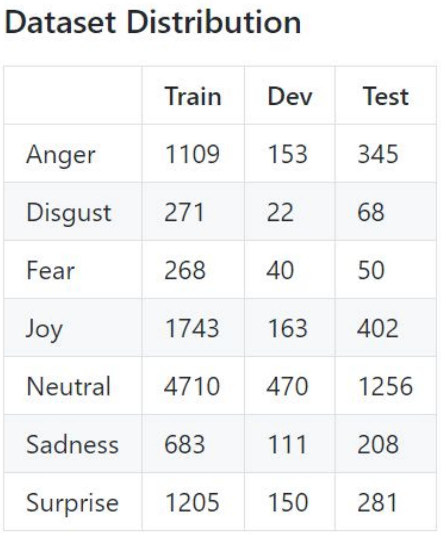
We used **MELD**: A Multimodal Dataset for Emotion Recognition in Conversation as the base dataset for our project. This dataset is available at MELD.

MELD has been created by enhancing and extending the **EmotionLines** dataset. MELD contains the same dialogue instances available in EmotionLines, but it also encompasses **audio** and **visual** modality along with the **text**. MELD has more than **1400 dialogues** and **13000 utterances** from the Friends TV series. Multiple speakers participated in the dialogues. Each utterance in dialogue has been labeled by any of these seven emotions -- **Anger**, **Disgust**, **Sadness**, **Joy**, **Neutral**, **Surprise**, and **Fear**. MELD also has sentiment (**positive**, **negative**, and **neutral**) annotation for each utterance.



Dataset Statistics

Statistics	Train	Dev	Test
# of modality	{a,v,t}	{a,v,t}	{a,v,t}
# of unique words	10,643	2,384	4,361
Avg. utterance length	8.03	7.99	8.28
Max. utterance length	69	37	45
Avg. # of emotions per dialogue	3.30	3.35	3.24
# of dialogues	1039	114	280
# of utterances	9989	1109	2610
# of speakers	260	47	100
# of emotion shift	4003	427	1003
Avg. duration of an utterance	3.59s	3.59s	3.58s

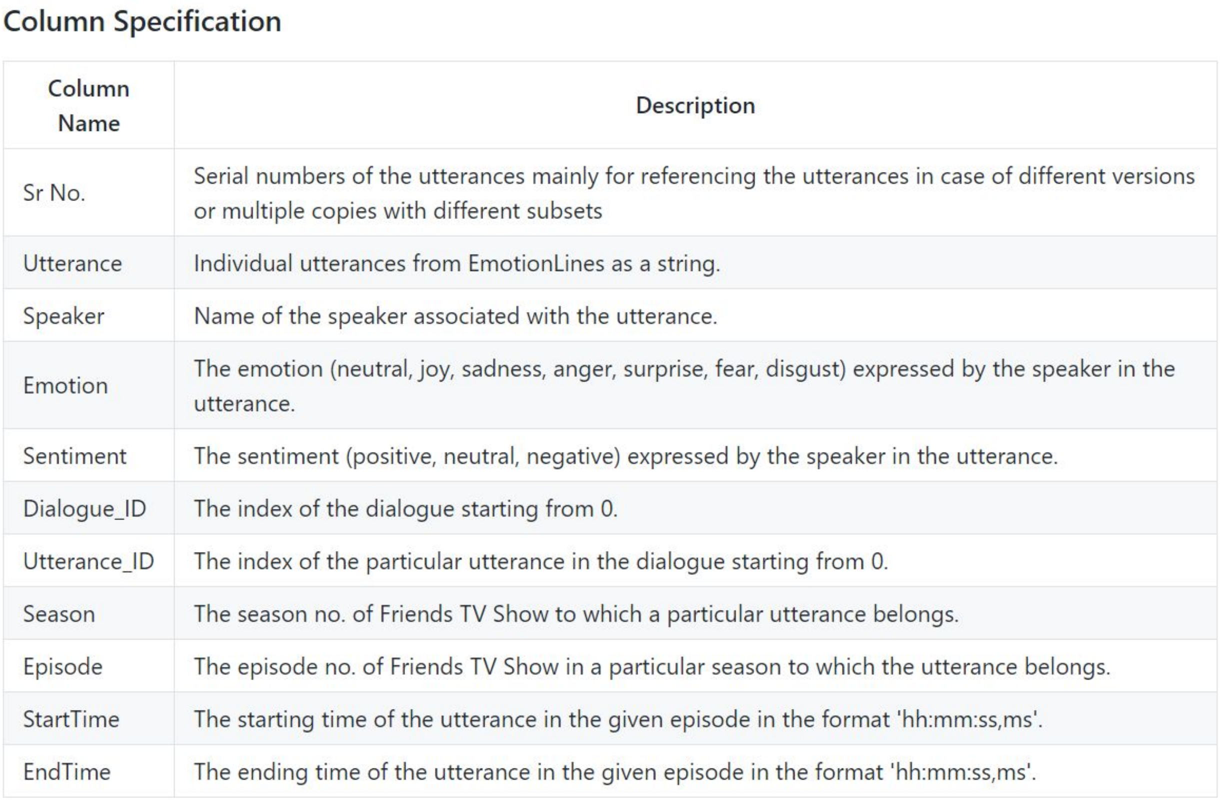


Dataset Distribution

	Train	Dev	Test
Anger	1109	153	345
Disgust	271	22	68
Fear	268	40	50
Joy	1743	163	402
Neutral	4710	470	1256
Sadness	683	111	208
Surprise	1205	150	281

Methodology:

The downloaded dataset contained text data in the below format. The dataset was divided into three parts: train, dev, and test. There were also video files present in mp4 format for all the dialogues. The audio used was extracted from these video files.
There were **9989** utterances in the **training set**, **2747** utterances in the **testing set**, and **1112** utterances in the **dev set**.



Column Specification

Column Name	Description
Sr No.	Serial numbers of the utterances mainly for referencing the utterances in case of different versions or multiple copies with different subsets
Utterance	Individual utterances from EmotionLines as a string.
Speaker	Name of the speaker associated with the utterance.
Emotion	The emotion (neutral, joy, sadness, anger, surprise, fear, disgust) expressed by the speaker in the utterance.
Sentiment	The sentiment (positive, neutral, negative) expressed by the speaker in the utterance.
Dialogue_ID	The index of the dialogue starting from 0.
Utterance_ID	The index of the particular utterance in the dialogue starting from 0.
Season	The season no. of Friends TV Show to which a particular utterance belongs.
Episode	The episode no. of Friends TV Show in a particular season to which the utterance belongs.
StartTime	The starting time of the utterance in the given episode in the format 'hh:mm:ss,ms'.
EndTime	The ending time of the utterance in the given episode in the format 'hh:mm:ss,ms'.

We followed the following steps in our methodology:

1. Data Preprocessing:

- Converted **Text Data** to **Pandas** Dataframe
- Removal of redundant columns such as (Speaker, Season_name, Episode, StartTime, EndTime)
- **Label Encoding** of Emotion and Sentiments.
- Organising the **Audio Data** to align with the **Text Data**.

2. Feature Extraction:

Text Features: We extracted **4 types** of text features from each utterance in the MELD dataset:-

- First we extracted all the **unigrams** that were present in an utterance
- Then we extracted all the **bigrams** that were present in each utterance
- We then used the **lexicons** to create various sentiment based features
- We also created a feature for the **sentiment** of each utterance that was given by creating a **label-encoding** for positive, negative and neutral

Audio Features: We used an open source software **openSmile** to extract **audio** features from the video of each utterance that was present in the dataset.

We then created a **feature vector** combining **Audio** and **Text** features, and then standardized them.

3. Feature Selection:

- **Feature vector dimension before feature selection:** 3168 features.
- **Anova:** We selected the best 500 features using **KBest** (anova).
- **Dimensionality Reduction using Principal Component Analysis:** We further applied **PCA**, on the selected features, and selected those principal components that retained **95%** of the information.
- **Feature vector dimension after feature selection:** 276 features.

4. Model Training:

We selected **SVM**, **Random Forest Classifier**, **MLP**, **AdaBoost**, **KNN** as the models for our project. We used **GridSearch** to find the optimal hyperparameters for each model. We then calculated the **accuracy**, **weighted F1-Score**, **weighted accuracy** for each model. The results from the models are present in the Results section.

Results:

After training all our models, and running them on the **test set**, we collected the **accuracy** scores, the **weighted-accuracy** scores and the **weighted F1-Scores** as the evaluation metrics. We also noted the differences between the evaluation metrics before and after including the **sentiment** of a given dialogue as a feature. We also collected the **emotion-wise** scores for each emotion, for the **SVM** model, which was our **best performing model**.

The respective scores are given below in the tables:

Table 1: Comparison of various models with various modalities.

(**SVM**: Support Vector Machines, **MLP**: MultiLayer Perceptron, **RF**: Random Forest, **AB**: AdaBoost, **KNN**: K Nearest Neighbours)(On test set)

Table 1: Comparison of various models with various modalities.

Models		Accuracy	Weighted F1-Score	Weighted Accuracy
SVM	text	77.24	83.93	**75.22**
	audio	48.54	64.58	20.82
	text+audio	**78.50**	**83.98**	60.65
MLP	text	70.95	70.96	44.23
	audio	34.67	34.96	17.96
	text+audio	75.17	76.12	44.88
RF	text	65.28	74.53	48.20
	audio	47.96	63.72	22.72
	text+audio	52.68	65.96	56.89
AB	text	48.12	64.97	48.12
	audio	48.12	64.97	48.12
	text+audio	59.61	67.31	42.67
KNN	text	62.49	69.55	57.24
	audio	39.38	42.87	35.27
	text+audio	47.85	56.22	43.58

Table 2: Model Performance with and without Sentiment as a feature (Weighted F1 Score on test set)

Model	With Sentiment	Without Sentiment
SVM	**83.98**	64.09
MLP	76.12	48.64
Random Forest	65.96	64.52
AdaBoost	67.31	**64.97**
KNN	56.22	52.83

We observe a **major increase** in the **performance** of the models after including **sentiment** as a feature.

Table 3: Label wise statistics for SVM, using both text and audio features

Emotion	Precision	Recall	F1-Score	Count
Anger	0.42	0.99	0.59	345
Disgust	0.01	0.02	0.01	68
Fear	0.02	0.02	0.02	50
Joy	0.77	0.99	0.87	402
Neutral	0.98	0.99	0.99	1256
Sadness	0.82	0.09	0.16	208
Surprise	0.02	0.01	0.01	281

File Structure:

- *lexiconFeatureVector.py* contains code for creating the **lexicon** features.
- *audio_feature_extraction.py* has code for extracting **audio features** from the the mp4 files of all the utterances.
- *text_feature_extraction.py* contains code to extract **text features** from the utterances.
- *model_train.py* has code for all the **models** that we used for **training** our feature vectors.
- *test.py* is used to create feature vectors for the **test** dataset and produce the **predicted outputs** using the models that we trained.
- The trained models are saved in the folder *models*

Notes:

a. The **code files** along with the **trained models** have been uploaded on **Google Classroom**.
b. The **dataset** link has been given in the **Dataset used** section.