Emotional tone recognition from speech and text:

a supervised machine learning and affective computing approach

by

Divya Vani Lakkireddy

B. Tech, Jawaharlal Nehru Technology University (JNTU), India, 2015

A REPORT

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2021

Approved by:

Dr. William Hsu

Major Professor

Copyright

Abstract

This report presents a system for classification of emotional tone in speech and text sequences using machine learning models ranging from a number of shallow atemporal models to recurrent deep learning. Identifying emotion from the speech is always a significant task. Emotion recognition experiment is carried out on speech features, text features from the speech transcriptions and combination setting of both speech and text features. We build a Long Short-Term Memory classifier to recognize emotion when a speech signal is given as input. The model is evaluated on the IEMOCAP, under multiple settings, namely, Audio-Only, Text-Only, and Audio + Text. For comparison, we have two approaches. For both, we extract eight features from the audio signal. In the first approach, the extracted features are used to train six machine learning classifiers, and in the second approach, a feedforward neural network and an LSTM-based classifier are used. The experiment is conducted in three experimental settings as Audio-Only, Text-Only, and Combined Setting where Audio features, Text features, and both are used for training the model respectively.

Keywords—speech emotion recognition, LSTM, machine learning, deep learning.

Table Of Contents

List Of Figures

List Of Tables

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. William Hsu, for his guidance in completing this project and his constant support during my graduate study at K-state. I would like to thank Dr. Torben Amtoft and Dr. Lior Shamir for serving as core members of my M.S. committee.

I would like to thank my parents Mr. Yogeswara Reddy Lakkireddy, Mrs. Narayanamma Lakkireddy and my brother Pavan Kumar Reddy Lakkireddy for their constant love and support without which I would have not achieved this.

Finally, I would like to thank my friends Sujith Gunturu, Niketa Penumajji, Dixit Kanchumarthy, Manoj Pulivarthy, Naveen Jonnalagadda, Rakshith Kumar and Rahul Harsha for their friendship, love, support and guidance throughout my master's Study. My Master's duration was all fun and interesting all because of their constant support and well wishes.

Chapter 1

Introduction

Communication is always key to human existence and there are many ambiguous situations where the expression can change its meaning based on the emotion expressed by the person. For example, “I don’t know what to do” this sentence can be a sad, excited, or happy form of the emotion and humans are able to resolve the ambiguity based on the speech, text, and visual domains.

Emotion Recognition in the current trend has many applications. This helps in Anger detection which can serve as a quality measurement in call centers and other voice portals. In civil aviation, emotion recognition can help in monitoring the stress of pilots which can help reduce aircraft accidents from occurring. In the video gaming industry, emotion recognition can help enhance the player experience by adjusting the game according to the emotion of the player. In the mental health care area, emotion recognition can be applied by assessing the subject’s psychiatric disorder. Emotion Recognition can also be used in conversational chatbots.

Speech emotion recognition (SER) is majorly divided into three main stages:Dataset selection, feature extraction and classification. During the feature extraction stage, a speech signal is converted to numerical values by front-end signal processing techniques. Then the extracted feature vectors which have a compact form captures the required information from the signal. In the back-end, an appropriate classifier, in our case a Long Short-Term Memory classifier is selected according to the task to be performed.

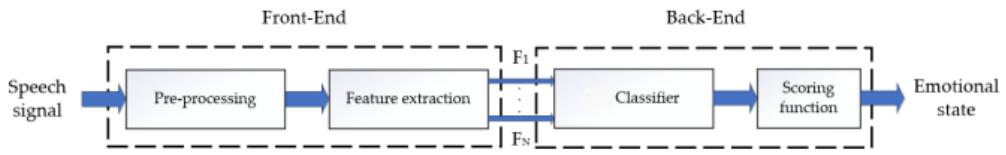

Figure 1: Schema of speech emotion recognition system

LSTM network is opted in my research as these networks can bridge the larger time intervals even if the input sequence is incompressible and has noise. LSTM has a gradient-based algorithm with constant error flow; they can truncate the gradient computations without affecting the long-term activations. Currently with advancements in deep learning, LSTM has become the major attention for the applications which involve time series occurrences. Speech processing and speech emotion recognition are such applications which involve time series events.

Emotion recognition experiment is carried out on speech features, text features from the speech transcriptions and combination setting of both speech and text features. We build an LSTM classifier to recognize emotion when a speech signal is given as input. The model is evaluated on the IEMOCAP, under multiple settings, namely, Audio-Only, Text-only and Audio + Text. Then a comparative study between the deep learning models and lighter machine learning models which are trained end to end is conducted. A preliminary data preprocessing, feature extraction is carried out. There are two approaches, one is machine learning classifiers which are Random Forests, Gradient Boosting, Support Vector Machines, Naïve-Bayes, and Logistic Regression and in the other approach we have Multi-Layer Perceptron and LSTM.

Chapter 2

Background and Related Work

The origins of Affective Computing were in the 1960s, when the ability of humans increased in transferring the thoughts to machines and vice-versa. This was at first referred to as "man-machine coupling" [16]. The initial approach relied on control theory, not programming. In Late 90's research became interested in human computer interaction and applications in reducing user frustration, enabling communication of user emotion, tools to enable social emotional skills, and developing infrastructure and applications to handle affective information [17]. One of the ultimate goals of affective computing is affective interactions, when emotional information is communicated by the user in a natural and comfortable way that is recognized by the computer and used to improve its interaction with the user [18] . This "Human-Computer Coupling" is known as affective computing in today's advancement.

In this section, we review some of the work that has been done in the field of speech emotion recognition. Majority of speech emotion recognitions use Hidden Markov Model previously. With the growing importance of Speech Emotion recognition in human-computer interaction there are many researchers working towards its advancement.

In 2018, Swain et al. [19] studied between 2000 and 2017 SER systems based on the database, feature extraction, and the classifiers. This research has focused on the database and the feature extraction. This research has worked on traditional machine learning methods for classification. Later Khalil et al. [20] proposed Speech Emotion Recognition using deep learning approaches like Deep neural network, convolutional neural network, recurrent neural network. Recently Anjali et al. [21] published a review as a summary of speech emotion detection methods which has discussion of multiple features in speech emotion recognition and reviewed methods from 2009 – 2018 has been provided but the depth analysis was not mentioned. In 2020, Basu et al. [22] a

review on speech emotion datasets and features, noise reduction was published. This has analysis of classification approaches of SVM and HMM. The drawback of the paper was the leak of modern methods analysis. Later Akcay et al. [23] published a survey with relatively comprehensive study on datasets, features, classifiers, and emotion models. However, there was no comparison study of different methods of machine learning and deep learning networks.

In addition to the other published reviews, my research is an experimental study of LSTM and includes a comparative study of lighter machine learning models and deep learning models. With the introduction of deep neural networks this domain has improved significantly as the prediction performance has increased. This work has implementation of deep learning-based models that are trained end-to-end, and lighter machine learning models trained over hand-crafted features.

Chapter 3

Methodology

This section discusses the various methodologies implemented in the experiment for Speech/Text Emotion Recognition.

3.1 Long Short-Term Memory (LSTM)

LSTMs [11] were introduced for long-range context capturing in sequences. LSTM has feedback connections which allow it to decide which information is important and which is not. It consists of a gating mechanism and has three types of gates: input, forget and output. These Gates are sigmoid activation functions. These give output values as 0 or 1.

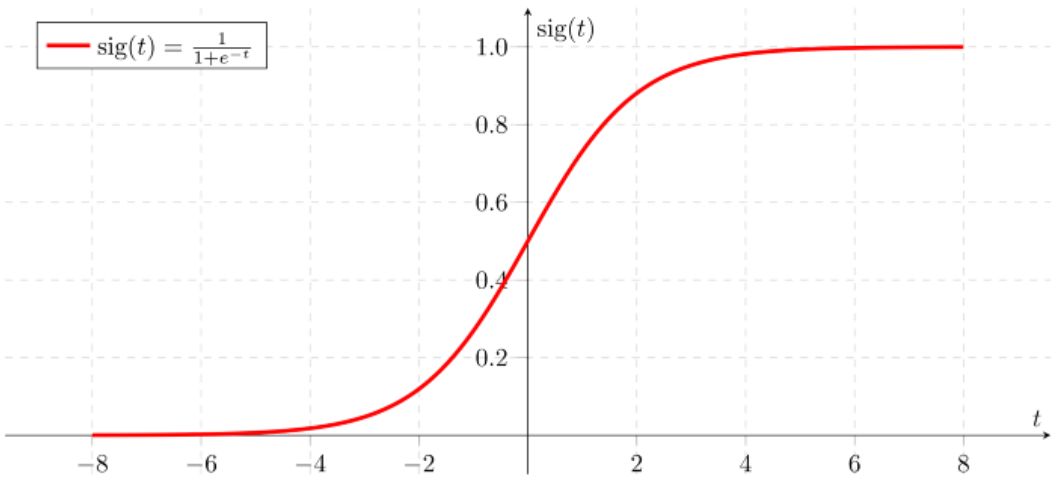

Figure 2: Sigmoid values between 0 and 1

This sigmoid function gives positive values and will be able to determine the feature to keep and the feature which needs to be discarded. “0” means the gates are blocking everything, “1” means gates are passing everything through it.

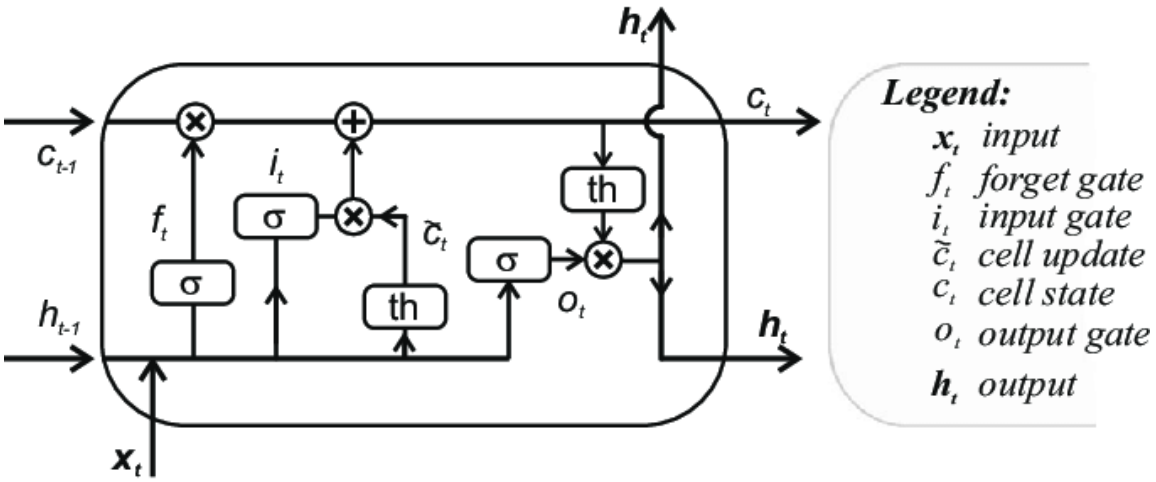

Figure 3: LSTM memory cell

The key to the LSTM cell is its cell state. This is a horizontal line as shown in Figure 3. In the first step of the LSTM, we decide which information needs to be discarded and which information needs to be kept from the cell state. Sigmoid layer which is known as forget gate layer takes this decision. This is based on h_{t-1} and x_t. This gives output between 0 and 1 for cell state C_{t-1}. If the value is 1 it means to keep the information and if it is 0 it means to completely discard the information.

$$f_t = \sigma\left(W_t . [h_{t-1}, x_t] + b_f\right) \qquad (1)$$

Next, we decide on what new information needs to be stored in the cell state. This has two parts. In the first part, the sigmoid layer which is the input gate layer (i_t) decides the values that need to be updated. Next, we have a tanh layer which creates the new value of the C_t. These two layers are used to create an update to the cell state.

$$i_t = \sigma\left(W_i . [h_{t-1}, x_t] + b_i\right) \qquad (2)$$

$$C_t = \tanh(W_c . [h_{t-1}, x_t] + b_C) \qquad (3)$$

$$C_t = f_t * C_{t-1} + i_t * C_t \qquad (4)$$

The old cell state C_{t-1} updates to new cell state C_t. By multiplying the old state by f_t, then adding $i_t * C_t$. The resultant will be the new value. At the end output is based on the cell state which will be a filtered version. The sigmoid layer decides the output for the cell state. Then the cell state is put through the tanh and multiplied by the output of the sigmoid gate.

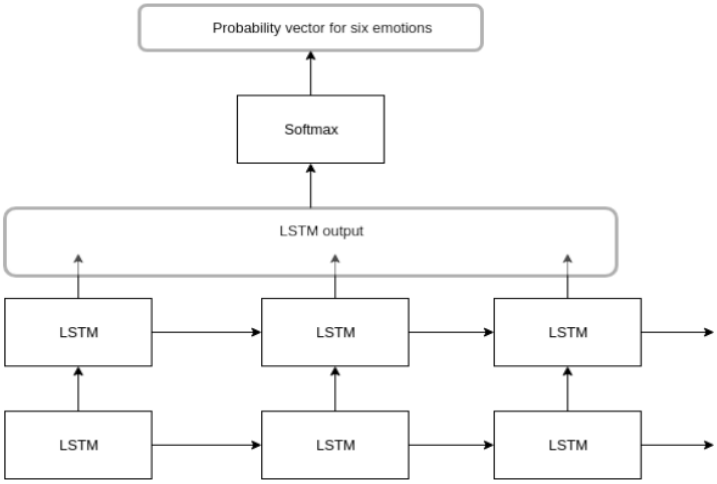

Figure 4: LSTM classifier

Above Figure 4 is the implementation of LSTM in this work. Feature vectors are fed as input to the network, output of the LSTM network is passed to a SoftMax layer which gives the probability scores for all the emotion classes. As we have feature vectors as input, we do not need the decoder network to transform hidden to output which reduces the network size. In order to regularize the hidden space, a dropout mechanism is used where a fraction of neurons are not used for final prediction. This improves the network and prevents overfitting. Dataset is split into train and test as 80% and 20% respectively. Same split is used for other models for fair comparison.

3.2 Other Models Used for Comparison

3.2.1 Multi-Layer Perceptron (MLP)

MLP is a class of feedforward neural networks. It consists of at least three nodes which are: an input, a hidden and an output layer. All the nodes are interleaved with a non-linear activation function to stabilize the network during training time. The expressive power increases with increase in the number of hidden layers.

3.2.2 Random Forest (RF)

Random forests are ensemble learners that operate by constructing multiple decision trees at training time and outputting the class that is the mode of the classes of the individual trees. It has two base working principles: 1) Each decision tree predicts using a random subset of features [6] 2) Each decision tree is trained with only a subset of training samples. This is known as bootstrap aggregating [7] Finally, a majority vote of all the decision trees is taken to predict the class of a given input.

3.2.3 Gradient Boosting (XGB)

XGB refers to extreme Gradient Boosting. It is an implementation of boosting that supports training the model in a fast and parallelized way. Boosting is another ensemble classifier combining several weak learners, typically decision trees. They are trained in a sequential manner, unlike RFs, using forward stagewise additive modeling.

3.2.4 Support Vector Machines (SVMs)

SVMs are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The main objective of SVM's is to find the hyperplane in the N-dimensional space which classifies the data point. [8] SVMs were originally introduced to

perform linear classification; however, they can efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

3.2.5 Multinomial Naive Bayes (MNB)

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Under multinomial settings, the feature vectors represent the frequencies with certain events being generated by a multinomial (p1, . . . , pn) where pi is the probability that event "i" occurs. MNB is very popular for document classification tasks in text [9] which too essentially is a multi-class classification problem.

3.2.6 Logistic Regression (LR)

LR is used for binary classification problems [10], that is, when we have only two labels. In this work, LR is implemented as one-vs-rest. six classifiers have been trained for each class and then we consider the class which is predicted with the highest probability.

3.3 Dataset

In this work, we use the IEMOCAP, [5] released in 2008 by researchers at the University of Southern California (USC). It is abbreviated as Interactive Emotional Dyadic Motion Capture. It contains five recorded sessions of conversations from ten speakers and amounts to nearly 12 hours of audio-visual information along with transcriptions. It has audiovisual data which includes video, speech, motion capture of the face and text transcriptions.

Figure 5: Recording session

Figure 6: Motion capture of the face

It has eight categorical emotion labels, which are anger, happiness, sadness, neutral, surprise, fear, frustration and excitement. The dataset is split into multiple utterances for each session and we further split each utterance file to obtain wav files for each sentence. This was done using the start timestamp and end timestamp provided for the transcribed sentences. This finally has a total of ~10K audio files which are used to extract features. This semi-natural database has naturally emitted emotions compared to other simulated databases. IEMOCAP, is a well-suited dataset for deep learning applications.

A preliminary frequency analysis on the dataset revealed that the dataset is not balanced. The under-represented emotions “fear” and “surprise” were unsampled. “Happy” and “excited”

classes were labeled as “happy” as they closely resemble each other. Data classified as “others” was discarded as these were examples corresponding to the ambiguous data. After applying these operations, the final dataset resulted in 7837 examples in total. Final distribution for each of the emotions is shown in Table I.

Table 1: Examples in each emotion

Class	**Count**
Angry	860
Happy	1309
Sad	2327
Fear	1007
Surprise	949
Neutral	1385
Total	**7837**

Chapter 4

Experiment

The first step involved in the experiment setting is the Data Preprocessing, Feature Extraction, Machine Learning Models, Deep Learning Models. This section gives a detailed description of all the steps involved in the experiment.

4.1 Modes of Experiment

Here, we describe the three settings which are conducted in the experiments:

a. Audio-Only: Only audio feature vectors are used to train all the classifiers.
b. Text-Only: Only text feature vectors are used to train all the classifiers.
c. Audio + Text: Concatenation of speech and text vectors into combined feature vectors trained on LSTM network.

4.2 Libraries used

Here we discuss the various libraries used for implementation of machine learning models and deep learning models.

a. librosa [12], a Python library, to process the audio files and extract features from them.
b. scikit-learn and xgboost [13] [14], the machine learning libraries for Python, to implement all the ML classifiers (RF, XGB, SVM, MNB, and LR) and the MLP.
c. PyTorch [15] to implement the LSTM classifiers described earlier.

4.3 Data Preprocessing

There are two types of data available in the IEMOCAP, for preprocessing: Audio and Text data.

4.3.1 Audio

Happy and excited classes are merged as they closely resemble each other. After the pre-analysis below are the emotion classes identified: Angry 860, Happy 1309, Sad 2327, Fear 1007, Surprise 949, Neutral 1385 which is a total of 7837 examples.

4.3.2 Text

The available transcriptions are normalized to lowercase and special symbols are removed. Extracted the labels from the transcripts and compiled the required data into csv. Built audio vectors from wav files and saved them to a pickle file. Once the labels have been extracted, compiled csv (df_iemocap.csv) is used to split the original wav files into multiple frames.

4.4 Feature Extraction

All the features are extracted using librosa library and respective Speech data files and text files are built with respective features.

4.4.1 Audio Features

To differentiate the actual emotion behind the utterance of a sentence we try to extract some features which help us to determine whether the speaker meant the statement to be anger, sadness or happiness. We discuss the five audio features which are extracted for the implementation of the speech emotion recognition system.

a) Pitch: Pitch is an important feature as the waveforms produced by the vocal cords change depending on the emotion (autocorrelation of center-clipped frames). The input signal y[n] is center-clipped which gives the resultant signal, $y_{clipped}[n]$.

$$y_{clipped}[n] = \begin{cases} y[n] - C_l, \text{ if } y[n] \geq C_l \\ 0, \text{ if } |y[n]| < C_l \\ y[n] + C_l, \text{ if } y[n] \leq C_l \end{cases} \quad (5)$$

C_l is nearly half the mean of the input signal and [·] implies the discrete nature of the input signal. Autocorrelation is calculated for the signal $y_{clipped}$, which is normalized further. By centerclipping the input signal more distinct autocorrelation peaks are obtained. This calculation of pitch using the librosa library is shown in Figure 7.

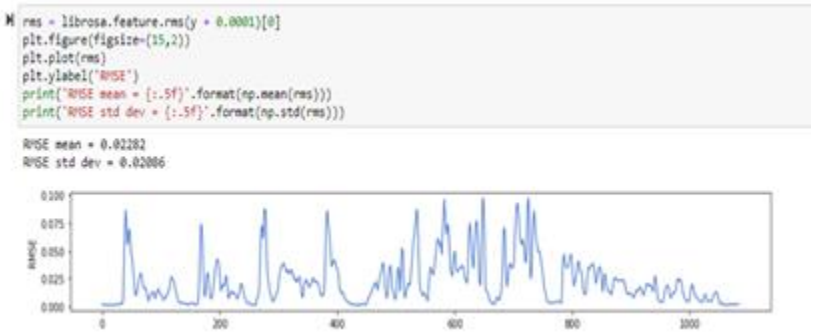

Figure 7: rms calculation

b) Harmonics: There exists excitation signals in the voice when in an emotional state of anger or stressful speech. Figure 8 shows the Harmonic of Angry and Sad(Red and Blue) audio signals. The harmonics for “angry” have a higher peak than that of the “sad” emotion. Figure 9 shows the Harmonics calculations using the librosa library. This is calculated using a median-based filter. The median filter is created for window size l

$$y[n] = \text{median}(x[n-k : n+k] | k = (l-1)/2) \quad (6)$$

When l is even, the median is obtained as the mean of two values of the sorted list. Then this is applied to S_h, the h−th frequency slice of a given spectrogram S, In order to get harmonic-enhanced spectrogram frequency slice H_h as

$$H_i = M(S_h, l_{harm}) \qquad (7)$$

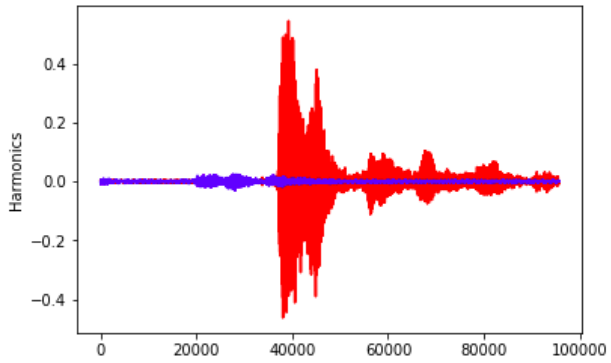

Figure 8: Harmonic of Angry and Sad (Red and Blue) audio signals

```
y_harmonic, y_percussive = librosa.effects.hpss(y)
np.mean(y_harmonic)
```

```
77]: -5.0849667e-06
```

Figure 9: Harmonics calculations using librosa

c) Speech Energy: The energy of the speech signal is related to loudness; hence this can be used to detect certain emotions. For this standard Root Mean Square is used.

$$E = \frac{1}{n}\sqrt{\sum_{i=1}^{n} y[i]^2} \qquad (8)$$

Below Figure 10 shows the difference in energy levels for anger and sadness. RMSE is calculated frame by frame with average and standard deviations as features.

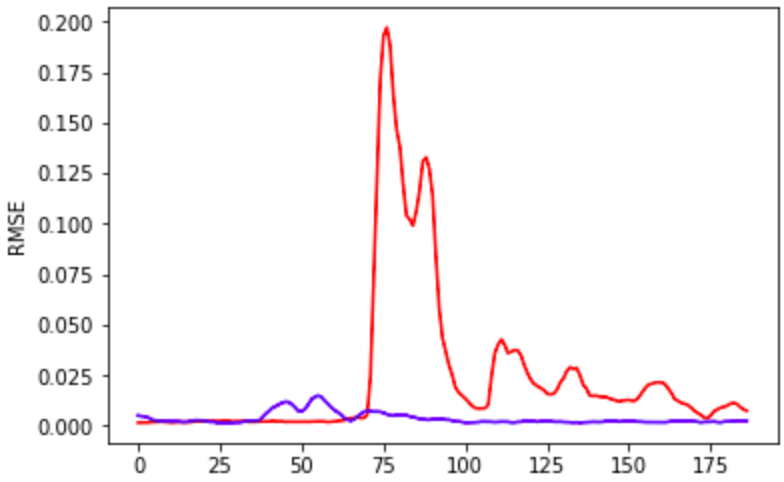

Figure 10: RMSE plots of angry and sad

d) Pause: This feature is used to represent the "silent" portion in the audio signal. This feature is equivalent to emotions. For example, one tends to speak faster when in excitement and speaks slower when in sorrow. Figure 11 shows the calculation using the rms value. This Pause is calculated using below equation

Pause = P r(y[n] < t) where t is threshold of 0.4 * E, E is RMSE. (9)

```
silence = 0
for e in rms:
    if e <= 0.4 * np.mean(rms):
        silence += 1
print(silence/float(len(rms)))
```

```
0.29704797047970477
```

Figure 11: Silence calculation

e) Central moments: Finally, we use the mean and standard deviation of the amplitude of the signal to incorporate a "summarized" information of the input. Figure 12 shows the central moments calculation using the mean and standard deviation.

```
autocorr = librosa.core.autocorrelate(y)
np.max(autocorr)
```

[78]: 530.6467406775207

```
cl = 0.45 * np.mean(abs(y))
center_clipped = []
for s in y:
    if s >= cl:
        center_clipped.append(s - cl)
    elif s <= -cl:
        center_clipped.append(s + cl)
    elif np.abs(s) < cl:
        center_clipped.append(0)
new_autocorr = librosa.core.autocorrelate(np.array(center_clipped))
np.max(new_autocorr)
```

[79]: 392.05612712951057

Figure 12: Calculation of central moments

4.4.2 Text Features

In this section we discuss the text features which are extracted for the implementation of the speech emotion recognition system.

Term Frequency-Inverse Document Frequency (TFIDF): It is a numerical statistic which shows the correlation between a word and a document in a collection. TFIDF is a product of Term frequency and Inverse document frequency. This has two parts:

i. Term Frequency: This denotes how many times a word/token occurs in a document. The simplest choice is to use the raw count of a token in a document (sentences, in our case).

ii. Inverse Document Frequency: This term is introduced to lessen the bias due to frequently occurring words in language such as “the”, “a”, and “an”.

Labels from transcriptions are extracted and other required data is compiled into a csv. Vectors are built from original wav files and are saved in the pickle file. Then 8-dimensional audio feature vectors for the audio are extracted, preprocessed and the data is prepared for the experiment. We then randomly split our dataset into a train (80%) and test (20%) set and train the LSTM classifier model in order to predict the emotion.

4.5 Evaluation Metrics

Accuracy, Precision, Recall and F1-Score are considered as the evaluation metrics for the performance analysis. Below Figure 13 shows the confusion matrix terminology with which the evaluation metrics are calculated. These metrics help in the performance evaluation of the model in predicting the emotional class correctly. Then, the Confusion matrix which gives the summary of the correct and incorrect predictions made by the classifier is implemented. This gives the count of each emotion class predicted.

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 13: Confusion matrix terminology

Chapter 5

Results

5.1 Accuracy

In this section we will discuss the results obtained for the audio experiment setting, text experiment setting and audio + text experiment setting. We will see the performance of all the models considered for comparison. Accuracy, Precision, Recall and F1 score for all the models is calculated and shown in the respective tables.

5.1.1 Audio Setting: Accuracy, Precision, Recall and F1-Score

Accuracy of the LSTM classifier for Audio-Only settings is 70.4%. Audio setting was then compared with lighter Machine learning classifiers: Random Forest, Gradient Boosting, Support Vector Machine, Multinomial Naive Bayes, Logistic Regression and MLP.

Model	Accuracy	Precision	Recall	F1 Score
LSTM	70.4	68.9	66.3	67.6
MLP	68.3	67.6	65.2	66.4
RF	68.2	67.2	66	66.6
XGB	67.1	67	66.7	66.8
SVM	65.2	64.8	63.9	64.3
MNB	65.7	64.9	63.1	64
LR	65	64.1	63.3	63.7

Table 2: Audio-Only setting evaluation metrics

The performance of LSTM model using speech segment based feature extraction achieves highest accuracy over all the other models. The results shown by the lighter machine learning models was also not behind the LSTM which was trained on the eight dimensional audio feature vector. The comparison results of all the models is given in the below Table 2.

5.1.2 Text Setting: Accuracy, Precision, Recall and F1-Score

Accuracy of the LSTM classifier for Text only settings is 73.9%. We observe that the performance of all the machine learning models and LSTM is similar. This could be because of the richness of the TFIDF(Term frequency-Inverse document frequency) vectors which are known to capture word-sentence correlation. The comparison results of all the models is given in the below Table 3.

Model	Accuracy	Precision	Recall	F1 Score
LSTM	73.9	74.2	71.6	72.8
MLP	72.8	71.8	66.2	68.9
RF	72.3	71.6	68.9	70.2
XGB	71.2	69.9	68.9	69.39
SVM	70.4	68.9	66.3	67.6
MNB	69.7	67.9	64.1	65.9
LR	68.5	64.4	63.7	64

Table 3: Text Only setting evaluation metrics

5.1.3 Audio + Text Setting: Accuracy, Precision, Recall and F1-Score

The combination of speech and text emotion recognition improves the performance of the models for all the metrics. This implies that there exists a strong correlation between text and speech features. Text features combined with speech features improved the emotion recognition accuracy. The best performing model LSTM is implemented for the combination setting of the speech and text features. There is a growth in the performance and the accuracy of the combined model is 78.9%. Table 4 gives all the evaluation metrics for the best performing model LSTM.

Combined Model	Accuracy	Precision	Recall	F1 Score
LSTM	78.9	78.3	77.6	77.9

Table 4: LSTM combined evaluation metrics

5.2 Confusion Matrix

Confusion matrix is a summary of the correct and incorrect predictions made by the classifier. This gives the count of each class predicted.

5.2.1 Audio-Only Setting

A look at the Figure 14 which is a confusion matrix reveals a more detailed analysis of the results. Detecting “neutral” emotion was the most difficult task for the model. Model struggled distinguishing between “angry” and “sad” and also between “neutral” and “sad”. The highest accuracy obtained for the audio feature based model is 70.4% and LSTM is the best performing model of all other models with respect to all the metrics. x- axis is the predicted label and the y-axis is the true label.

Figure 14: Confusion Matrix of Audio Only Setting

5.2.2 Text-Only Setting

Our Text-based models fairly distinguish the six emotions which are shown in Figure 15. The highest accuracy obtained for the text feature based model is 73.9% and LSTM is the best performing model of all other models with respect to all the metrics.

Figure 15: Confusion Matrix of Text Only Setting

5.2.3 Audio+Text Setting

The combination of speech and text features in the best performing LSTM model has produced highest accuracy and helped to resolve the ambiguity observed for modality-specific models. By analyzing Figure 16 we can conclude that Textual features helped in classification of "angry" and "happy" classes whereas the audio features helped in detecting "sad" better. Accuracy of the combined setting model is 78.9%.

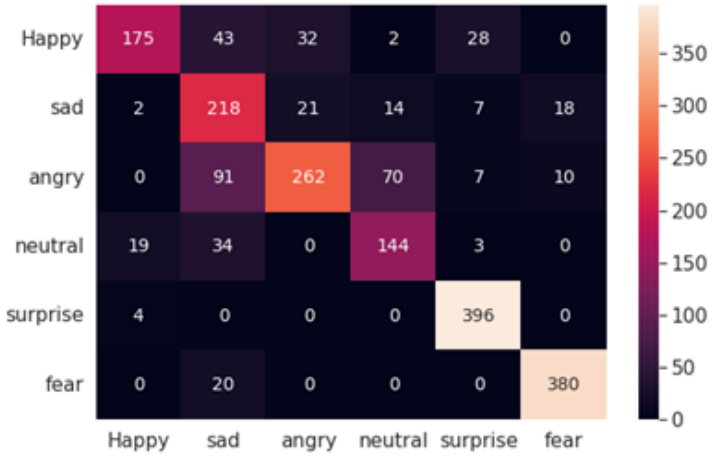

Figure 16: Confusion Matrix of Combined Setting

Chapter 6

Conclusion & Future Work

This chapter draws conclusions and provides glimpses of possible future work.

6.1 Conclusion

In this paper emotion recognition is performed on IEMOCAP, a dataset using data from speech, text, and motion capture over various classification models. The LSTM model has achieved robust performance and performed better over the comparative machine learning models and other deep learning models. LSTM network is proved as an effective model in the time series data because of its pattern memorizing capacity. Our model is trained to predict six emotion classes unlike the previous works where only four emotion classes are predicted.

Speech emotion recognition accuracy is improved by integrating Speech and text feature vectors trained over the LSTM network. The model has to be trained several times in order to get similar results throughout and accuracy of 78.9% is obtained which out-performs the previously obtained research results on the same dataset and similar model implementations which had accuracy of 75.49%. For future research a strategy can be implemented in order to get high accuracy which will enable a benchmark performance compared to other existing emotion recognition networks.

6.2 Future Work

The dataset used, IEMOCAP, has only ten actors recordings which produce multiple utterance-level speeches, but this has limitations for speaker-independent classification as it limits the generalization of training the model. Though the achieved accuracy is higher than existing research models but can be further improved in order to be incorporated in the real time applications. There are furthermore audio features which could be taken into consideration for future research like the Mel-Frequency Cepstral Coefficients (MFCC), Spectral Roll-off and time domain features like Zero Crossing rate.

By applying more fusion methods the combination of speech and time vectors models would produce effective results. For future work, large amounts of data could be used to see interesting results of Machine learning and deep learning models. Usage of large amounts of data would improve the autoencoder reconstruction effect and would result in robust feature extraction results. Optimized autoencoder models would generate more features which would further provide more reliable feature extraction architecture for future research.

The performance can be further improved at the cost of more process and exponential memory requirements as these networks require large training samples in order to tune the large number of variables.

There is a scope of conducting the research over various other datasets available to check the consistency of the result. In future other solutions for feature extraction and prediction algorithms can be explored. This research work has the ability to detect emotions with speech and text features as input. In the future there is a scope of developing a model capable of detecting emotions from video, images along with the speech and text, which will help in much clearer emotion recognition applications.

Bibliography

[1] K. Han, D. Yu, and I. Tashev, “Speech emotion recognition using deep neural network and extreme learning machine,” in Fifteenth annual conference of the international speech communication association, 2014.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Y. Kim, H. Lee, and E. M. Provost, “Deep learning for robust feature generation in audiovisual emotion recognition,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3687–3691, IEEE, 2013.

[3] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, “Tensor fusion network for multimodal sentiment analysis,” arXiv preprint arXiv:1707.07250, 2017.

[4] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency, “Efficient low-rank multi-modal fusion with modality-specific factors,” arXiv preprint arXiv:1806.00064, 2018.

[5] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” Language resources and evaluation, vol. 42, no. 4, p. 335, 2008

[6] Y. Amit, D. Geman, and K. Wilder, “Joint induction of shape features and tree classifiers,” IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 11, pp. 1300– 1305, 1997.

[7] L. Breiman, “Bagging predictors,” Machine learning, vol. 24, no. 2, pp. 123–140, 1996.

[8] C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[9] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, “Multinomial naive bayes for text categorization revisited,” in Australasian Joint Conference on Artificial Intelligence, pp. 488–499, Springer, 2004

[10] G. King and L. Zeng, “Logistic regression in rare events data,” Political analysis, vol. 9, no. 2, pp. 137–163, 2001.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in Proceedings of the 14th python in science conference, pp. 18–25, 2015.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in python," No. Oct, pp. 2825–2830, 2011.

[14] T. Chen, "Scalable, portable and distributed gradient boosting (gbdt, gbrt or gbm) library, for python, r, java, scala, c++ and more. runs on single machine, hadoop, spark, flink and dataflow," 2014

[15] Facebook, "Pytorch," 2017.

[16] RM Page. Man-machine coupling-2012 ad. Proceedings of the IRE, 50(5):613–614, 1962.

[17] Rosalind W Picard. Affective computing for hci. In HCI (1), pages 829–833, 1999.

[18] Carson Reynolds and Rosalind W Picard. Designing for affective interactions. In Proceedings from the 9th International Conference on Human-Computer Interaction, page 6, 2001

[19] Swain, M.; Routray, A.; Kabisatpathy, P. Databases, features and classifiers for speech emotion recognition: A review. Int. J. Speech Technol. 2018, 21, 93–120, doi.org/10.1007/s10772-018-9491-z.

[20] Khalil, R.A.; Jones, E.; Babar, M.I.; Jan, T.; Zafar, M.H.; Alhussain, T. Speech Emotion Recognition Using Deep Learning Techniques: A Review. IEEE Access 2019, 7, 117327–117345.

[21] Tripathi, A.; Singh, U.; Bansal, G.; Gupta, R.; Singh, A.K. A Review on Emotion Detection and Classification using Speech. In Proceedings of the International Conference on Innovative Computing and Communications (ICICC), Online, 15 May 2020.

[22] Basu, S.; Chakraborty, J.; Bag, A.; Aftabuddin, M. A Review on Emotion Recognition using Speech. In Proceedings of the International Conference on Inventive Communication and Computational Technologies (ICICCT 2017), Coimbatore, India, 10–11 March 2017.

[23] Akçay, Mehmet, B.; O ˘guz, K. Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers. Speech Commun. 2020, 166, 56–76.