SPEECH EMOTION RECOGNITION IN CONVERSATIONS USING GRAPH CONVOLUTIONAL NETWORKS

DEEKSHA CHANDOLA

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO
MARCH, 2024

Abstract

Speech emotion recognition (SER) is the task of automatically recognizing emotions expressed in spoken language. Current approaches focus on analyzing isolated speech segments to identify a speaker's emotional state. That being said, models based on text-based emotion recognition methods are considering conversational context and are moving towards emotion recognition in conversation (ERC). With the availability of multimodal datasets, ERC can be extended to non-text modalities as well. Building on these advances, in this thesis, we propose SERC-GCN, a method for speech emotion recognition in conversation (SERC) that predicts a speaker's emotional state by incorporating conversational context, specifically speaker interactions, and temporal dependencies between utterances. SERC-GCN is a two-stage method. In the first stage, emotional features of utterance-level speech signals are extracted using a graph-based neural network. Here each individual speech utterance is transformed into a cyclic graph. These graphs are then processed by a two layered GCN architecture followed by a pooling layer to extract utterance-specific emotional features. In the second stage, these features are used to form conversation graphs that are used to train a graph convolutional network to perform SERC. We empirically evaluate the effectiveness of SERC-GCN on two benchmark dataset; IEMOCAP and MELD. Results show that SERC-GCN outperforms existing baseline approaches on these datasets.

Acknowledgements

I would like to express my gratitude towards my supervisor, Prof. Michael Jenkin, for his unwavering support, kindness, and guidance throughout my grad school journey, from day one until today. His knowledge and experience always motivated me to work towards my goal with enthusiasm. Without his valuable feedback and assistance, the completion of this thesis would not have been possible. I would also like to extend my thanks to my committee member, Prof. Manos Papagelis, for his time and valuable feedback on this work. Their support has been instrumental in making my thesis sound and complete. I want to acknowledge my lab mate and best friend, Enas Tarawneh, for her unwavering support and mentorship in this work. Without her guidance and encouragement, the completion of this work would not have been possible. Lastly, I want to express my heartfelt gratitude towards my parents and sister for their constant motivation, and my husband and son, Dev, for being the pillars of strength in making my grad school journey possible.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

HCI Human Computer Interaction

ERC Emotion Recognition in Conversation

HRI Human–Robot Interaction

SER Speech Emotion Recognition

SERC Speech Emotion Recognition In Conversation

ESR Emotion Speech Recognition

GCN Graph Convolutional Network

GRU Gated Recurrent Unit

CNN Convolutional Neural Network

RNN Recurrent Neural Network

GPU Graphics Processing Unit

HMM Hidden Markov Model

LLD Low Level Descriptors

NLP Natural Language Processing

GNN Graph Neural Network

LSTM Long Short-Term Memory

BERT Bidirectional Encoder Representations from Transformers

LLM Large Language Model

MFCC Melfrequency cepstral coefficients

Chapter 1

Introduction

Conversations are the most basic interactive medium of communication. We engage in conversations almost every day of our lives. Messages in conversations may be task-oriented, or just general chit-chat, but they naturally convey emotion within the conversation. How can we enable machines to understand this emotional context? With the availability of large online datasets from social media websites (such as Facebook, Instagram, YouTube, Reddit, and many more), and the existence of conversational agents such as Amazon Alexa, Apple Siri, and Samsung S Voice, the development of emotion-aware dialogue systems have attracted considerable attention from the research community. The availability of a large number of conversation datasets (e.g.,[1], [2], [3]) supports this research as well. The aim of emotion analysis is to make a machine aware of the feelings and emotions of the person participating in the conversation. The challenge in this is that it is not easy to enable a machine to analyze the emotions in conversations since people often communicate emotion in very subtle ways relying on context and commonsense knowledge to encode emotion [4], which can be difficult to be identified by a machine.

To help introduce the task, consider the dialogue shown in Figure 1.1. This shows a sequence of only

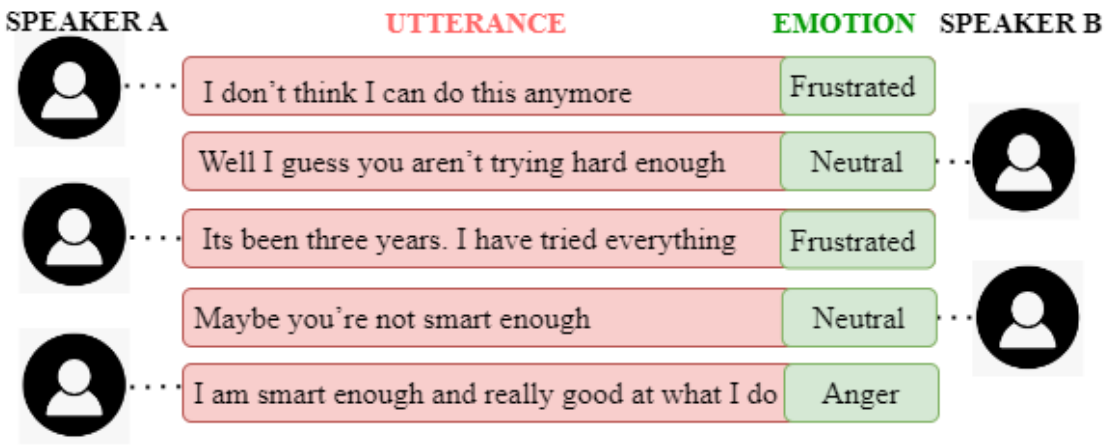


Figure 1.1: A turn-taking dialogue between two speakers A and B.

utterances between speakers A and B and the emotions associated with each utterance. In order to understand the conversation it is important to know the context because a conversation is a dynamic interaction in which each speaker conveys emotional messages. Having contextual information and commonsense knowledge can be very helpful in dialogue systems [5]. This context includes the order of utterances and the speaker associated with each utterance. We will return to this observation shortly.

Emotions expressed by a human can be captured and assessed by a machine in a variety of ways. Humans have access to a number of perceptual systems that can be leveraged to understand the emotion portrayed by the speaker including through vision and the processing of facial expressions (e.g., [6], [7]), hand gestures[8], text [9], and through audio and the acoustic nature of speech [10]. Understanding how a person is feeling over only a very short period of time helps to make emotion recognition a hard problem. But it was also the predominant approach taken in the early literature. An alternative approach, and the approach taken here, is to seek to estimate emotional context within a conversation to leverage the context and temporal nature of emotion recognition.

Emotion recognition in conversation (ERC) is the task of recognizing a speaker's emotion from an utterance in a conversation [11]. ERC has promising applications in many fields, such as recommendation systems, human-machine interaction, and medical care. Given a sequence of utterances in a conversation, as

shown in Figure 1.1, along with speaker information of each constituent utterance, the ERC task is typically formulated as the process of identifying the emotion of each utterance in the conversation from one of several pre-defined emotion classes.

There has been a long history of research in ERC using text data alone, but now with the availability of multimodal conversational datasets the ERC task can be expressed in terms of a range of different modalities including visual, text and audio cues, and their integration. Emotion analysis through text data alone can be very difficult. The limited nature of text can make emotion recognition ambiguous. For example, some individuals are more sarcastic than others which can be reflected in the voice but difficult to recognize when presented as text. For a sarcastic individual, the intent of certain words varies depending on if they are being sarcastic. Consider this example between person A and B. A utters "The party order has been cancelled.", B then utters "This is great!!". If B is a sarcastic person, then their response expression shows negative emotion. The word great is being expressed sarcastically. On the other hand, B's response, "This is great", might not be taken sarcastically if B is impacted positively by the cancellation. In this and similar situations, incorporating the nature of an audio utterance could be beneficial to understanding the tone of the utterance. In this work, we focus on audio (speech signals) as speech is an efficient and essential bridge in human communication [12].

Speech is a complex signal that contains information about the message, the speaker, the speaker's gender, intent, and emotions combined within it. Speech Emotion Recognition (SER) attempts to identify and detect human emotions by capturing and characterizing emotion attributes in speech utterances. Understanding the emotional state of the speech signal has also become a major research topic in Human-Computer Interaction (HCI), where it can enhance the emotional intelligence of the machine and makes the conversation more natural, in sync with the topic of the conversation, and more harmonious [13].

Beyond the modalities used to encode each speech utterance as described in [11] there exists a number of factors that influence the emotional structure of conversation including topic of conversation, the speaker's

personality, argument logic, intent, viewpoint and speaker's emotional state. Previous emotion recognition systems typically ignore conversation specific factors in their analysis. However, this contextual information plays an important role in emotion recognition. In this work, I focus on incorporating contextual information and speaker dependency in speech utterances in modelling Speech Emotion Recognition in Conversation (SERC). It is expected that the basic approach considered here could be adapted to a range of other properties associated with the conversation (e.g. text, facial gestures, etc.). The main research question I am exploring is "Can we enhance existing SER approaches through the use of conversational context"?

1.1 Motivation

Emotion Recognition in Conversations (ERC) and speech emotion recognition in conversations (SERC) in particular, is becoming an increasingly popular component in developing sympathetic human-machine interaction as it can be used to tailor an interaction to the perceived emotional state of the human. ERC is an active research topic due to its various applications, perhaps most especially in dialogue systems for human-computer interaction. The goal of human–computer interaction is not only to create a more effective and natural communication interface between people and computers, but also to create an attentive aesthetic design [14], provide a pleasant user experience [15], help in human development [16], and enhance online learning [17], among other goals. Since emotions form an integral part of human interactions, they have naturally become an important aspect of the development of effective HCI where conversational bots understand users' emotions and sentiments to generate emotionally coherent and empathetic responses [18].

A practical example of the need for speech emotion understanding and using this approach in interaction is shown in Figure 1.2. Sentrybot is a social robot designed to provide physical security to a facility and interact with visitors as it monitors the environment and reports on anomalies. With user interaction in the environment, Sentrybot could be used to assist in the de-escalation of civilian situations that could lead to open conflict, if left unchecked [19]. In conflict situations to achieve de-escalation is quite a complex

Figure 1.2: The Sentrybot showing its sentiment-informed avatar interface. Details on the robot can be found in [19].

task. This task requires a lot of different components to work together and is a collaborative exercise. It requires the system to monitor the emotional state of the human user, and also to express sentiment and conversational output in order to engage with users in socially effective ways. By generating personalized and natural responses, the system might be used to establish trust that can help to de-escalate a situation or a conflict.

For SER and SERC many different computational approaches have been proposed for emotion recognition, but in general, they have not considered the issue of speaker dependency even though the nature of speech is intrinsically speaker dependent as human speech production is the result of controlled anatomical movements of the lungs, trachea, larynx, pharyngeal cavity, oral cavity, and nasal cavity [20]. In the thesis, I incorporate contextual information and speaker dependency when modelling conversations.

1.2 Current approach

Assigning an emotional label to a given utterance involves developing a computational structure that can be used to do this assignment. Early work in this task relied on statistical models while most modern approaches are AI based. Current approaches to SER and SERC focus on analyzing isolated speech segments (i.e., an individual utterance considered in isolation) to identify a speaker's emotional state [21]. A major challenge in SER is that emotion-relevant information is scattered throughout different parts of the conversation. It is not easy to capture this distributed information. An efficient modeling capacity for long-term context is needed [22]. Speech emotion recognition in conversation attempts to identify and detect human emotions by capturing and characterizing emotion attributes in speech utterances within the conversational context rather than in isolated utterances [23]. The SERC problem involves two challenging tasks: (*i*) extracting utterance-level features, and (*ii*) modeling utterance dependencies in a conversation related to context, temporal and speaker dynamics [11]. Both parts are essential for effective emotion recognition.

1.3 Current limitations

Many, audio-based SER methods extract low-level descriptors (LLDs) from raw audio signals [24]. These are used as input for deep-learning models such as long-short-term memory networks (LSTM) [25], and convolutional neural networks (CNN) [26] to extract utterance-level features. Based on the network architecture nature, mostly deep learning SER models require equal-length input for all utterances during training. This can be achieved by padding the shorter utterances and truncating the longer utterances but it can lead to missing audio frames and noisy information [22]. As proposed by [27], adopting repeat frame padding or cycle padding can help to overcome this problem. But still the models do not preserve long-term emotional dependency on their own. For example, LSTMs model are able to capture the temporal nature of emotional speech dynamics [28], but this approach if incorporates various factors that influence the emotional structure

of conversation leads to a complex architecture with millions of parameters to learn which is computationally very expensive. Additionally, CNN-based approaches often extract emotional features from speech spectrograms for emotion recognition task [29] which is not an efficient approach as spectrograms do not explicitly model speech dynamics.

More recently, for speech emotion recognition in the literature the performance of LSTM-based and CNN-based methods have been superseded by the performance of graph convolutional network (GCN)-based models [30]. These models are more scalable, efficient, and compact and can represent complex relations and interdependencies in audio data. For example, a graph-based SER model is proposed by [31], which transforms each speech sample into a cyclic or line graph structure. This transformation simplifies the convolution operation and results in a more efficient, lightweight model with fewer parameters to train. Some other graph based approaches have been used [22, 32], and have shown state-of-the-art results. These models are capable of retaining emotional information and can represent complex dependencies in utterances. Motivated by this, this work adopts a graph-based approach for utterance-level feature extraction, similar to the one proposed by [31].

Modeling speaker, contextual and temporal dependencies is an active research topic in emotion recognition in conversation (ERC) [33, 34, 23]. Vareity of sequential models including GRU [35], BLSTM [36] , RNN [37], ResNet [38] and transformers [23] have been explored for modeling contextual dependency. However, these models do not include speaker and temporal dependencies which are essential for emotion recognition task. With the use of GCN-based models, incorporating context, speaker and temporal dependencies [33, 34] have achieved promising results for ERC task. For instance, Ghosal et al. [33] model a conversation as a directed relational homogeneous graph. In this graph, the nodes represent utterances and the edges represent a dependency between the two speakers of those utterances along with their relative order in the conversation. Unfortunately, when modeled as a complete graph, this approach results in a highly complicated graph structure. In addition, with a large number of speakers, the speaker dependencies are not fully encoded using this graph model. To address this problem Lian et al. [34] propose the use of heterogeneous graphs. In

the graph the nodes represent both utterances and speakers, and edges are created between immediate past and future utterances. Also, each speaker node is linked to the speakers associated with these utterances. This graph architecture can model more complex dependencies.

1.4 Our approach

Our approach to addressing the SERC problem aims at adapting the current state-of-the-art GNN-based ERC model [33], originally proposed for text-only input, to spoken dialogues, and in our case using audio data. Also, at the same time address the graph complexity limitation of the model. To address the graph complexity issue which affects learning meaningful speaker dependencies, the key idea is to adapt *graph sparsification* [39]. Sparsification is done by only considering past and future edges that occur within a small temporal window (recency) when constructing a relational conversation graph. Graph sparsification effectively eliminates a large number of edges from the original complete graph, while maintaining the most recent ones. In addition, our graph sparsification preserves a speaker's relation to its own utterances by maintaining same-speaker edges in the graph. As a result, we not only reduce the graph complexity but also manage to model *recency* and *same-speaker dependency* (or *self-dependency*), which are both important contributing factors in emotion recognition [33, 35, 40].

1.5 Applications

As touched upon earlier, there exists a range of application domains where having the ability to perceive and respond to human emotional state is regarded as an extremely desirable feature. Currently, emotion recognition systems are used in a wide variety of applications. For example, anger detection can serve as a quality measurement for voice portals or call centres [41]. Based on the situation and estimating the emotional state of the customer it is possible to adapt the provided services to the caller's estimated emotional state.

In the area of mental health care, a chatbot with emotion recognition capability is suggested to perform psychiatric counselling services [42]. It can also be used for services like therapy to comprehend patients effectively. SER and SERC can be used to provide insights in the underlying emotional state which are hidden in general see [43] and [44]. Anger detection is also useful in the development of social robots that interact with the general public and may be required to perform in situations within which upset individuals might be encountered. Another application domain is vehicle driving system benefit greatly from Speech Emotion recognition systems. SER provides an important tool towards enhancing vehicle safety [45]. The SER understands the emotions of the driver and guides against crashes and other road disasters. In civil aviation, monitoring the stress of aircraft pilots can help reduce the rate of aircraft accidents [46].

Researchers who seek to enhance a players' experiences with video games and to keep them motivated, can incorporate an emotion recognition module into their products. [47] describes a multi-modal emotion recognition system using audio and visual cues to provide a cloud-based gaming experience through emotion-aware screen effects. It can also be integrated into interactive movies. Here, the emotions of the viewers are gauged which can enable the producer of the movie to channel, or review such movies to end differently or maintain their ending as planned [48]. Another practical application of speech emotion recognition exists in online discussions and tutoring systems. Comprehension of the emotional state of the student can assist the tool to decide the technique or methodology to be adopted to present the remaining part of the lesson [45].

Perhaps the most general application for emotion recognition is in conversational chatbots, where emotions play a critical role in providing a better conversation [49]. Many people use some intelligent voice assistant device such as Siri (Apple), Google now, or S-Voice (Samsung). The basic concept of these services is that they respond to the users' inputs, such as queries of voice or text, and recommend useful information to the user based on the query. Incorporating emotional intelligence into these conversation systems can substantially improve their usability and promote customer satisfaction.

1.6 Contributions

The contributions of the thesis include:

- We propose SERC-GCN (Speech Emotion Recognition in Conversation using GCN), a two-stage graph-based deep learning model for the SERC problem [39]. SERC-GCN predicts a speaker's emotional state by incorporating the *conversational context*, *speaker interactions*, and *utterance temporal dependencies* in the model. The model architecture uses a two stage graph-based approach for SERC. In the first stage, the speech signal of each utterance is represented as a cyclic graph. These graphs are then processed by a two layered GCN architecture followed by a pooling layer to extract utterance-specific emotional features. In the second stage, these features are used to initialize the nodes of the conversation graph and a small temporal window is selected to form relational graphs of each conversation that are used to train a graph neural network. Our proposed model captures the conversation emotional dynamics through incorporating context, speaker dynamics, and utterance temporal dependencies.

- The relational graph-based model follows the approach adapted in the text-based GCN model described in [33], where our model takes into account speaker dependency and is order aware. This work is a step towards incorporating multimodal information into DialogueGCN model [33] as throughout this process we use audio utterances rather than text for the analysis.

- By taking into account speaker information/dependency and the relative position of the utterances in a conversation, our approach improves context understanding for utterance-level emotion detection in conversations and can be used to help create a richer context for relevant information to perform emotion recognition in conversations.

- We empirically evaluate the performance of our model on two benchmark datasets: IEMOCAP [2] and MELD [1]. The IEMOCAP dataset is a dyadic conversation dataset where the utterances in the conversations are labelled with the desired emotion. IEMOCAP was created using actors who

present their utterances using an acted emotional state. MELD is a multi-speaker dataset where the conversations take place between more than two people. MELD is based on conversation from the Friends TV show.

- For the IEMOCAP dataset, we develop two different utterance-only SER models. Set-1 is trained on the emotion categories: *happy*, *sad*, *neutral*, *angry* and *excited*. For this dataset our SERC-GCN model outperforms standard baselines. Specifically, it outperforms the second best model, Graph-SAGE, by 2.17% and 1.73% for weighted accuracy (WA) and unweighted accuracy (UA), respectively. SERC-GCN also outperforms Compact-SER [31], which is the underlying method used in the context-free stage of SERC-GCN. We also compare our model with other baselines of utterance-only SER models trained on a different set of emotion categories Set-2 from IEMOCAP: *angry*, *frustration*, *happy*, *neutral*, and *sad*. On this dataset our model shows an improvement of 1.2% in terms of macro F1 score as compared to transformer which shows the second best score. Comparing with conversational SER models for the emotion categories on Set-2 SERC-GCN outperforms all other SERC baselines. There is an improvement of 3.4% over the ResNet+Transfomer model [23] which is the second-best approach.

- For the MELD dataset, comparing our approach with the conversational SER baseline models for the emotion categories: *angry*, *disgust*, *sadness*, *joy*, *surprise*, *fear*, and *neutral* our model outperforms the all other models with an improvement of 5.9% over the graph-attention model [34] which is the second best approach for the dataset.

- This work has been accepted for presentation in the IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP 2024) and will be presented in Coex, South Korea on April 14-19, 2024 [39].

1.7 Thesis outline

The rest of the thesis is organized as follows. In Chapter 2 we look at related work in emotion recognition in conversation and introduce the technical problem of interest in speech emotion recognition. Chapter 3 formally introduces the task of speech emotion recognition in conversations and describes the training of Stage One and Stage Two methodology. An extensive set of experiments with two benchmark datasets that evaluate the performance of our approach is presented in Chapter 4, and finally, we conclude the thesis and discuss future work in Chapter 5.

Chapter 2

Background

2.1 Introduction

The current pandemic has highlighted the need for interactive sentiment-aware software systems, especially socially aware robot systems. Due to the labor shortages in the service sector and public health concerns around transmission, there has been a growing trend in deploying autonomous systems in various traditional roles such as server robots in restaurants, companion robots in long-term care homes, and security robots in public spaces, to name a few examples. To be successful, social robots should be able to communicate with a wide range of individuals under a variety of interaction scenarios [19]. Understanding and being aware by the emotion being expressed by an individual is key in human-human interaction [50], and is likely to be critical for effective human-robot interaction as well [51].

Humans express emotions in multidimensional ways, including through facial expressions, body language, hand gestures, the acoustic nature of speech, and text utterances. Emotion recognition is essential for effective

human-to-human communication, for the development of neutral human-computer interaction generally, and for the development of autonomous robots that interact with humans. Many researchers have adopted an interdisciplinary approach to recognizing emotions in conversations that uses advanced AI techniques and emotion models adapted from psychology. Machine learning methods can detect emotions by learning the features of emotions expressed through different modalities including text, speech and facial data, along with body language used including expressions, hand gestures and body movements. Psychology provides the emotion categorization theories that are needed to annotate and classify the data. Efficient and effective emotion recognition systems have been obtained by integrating a machine learning (ML)-based approach and the knowledge from other domains as illustrated in Figure 2.1.

Emotion recognition allows us to adapt the behavior of robots and other interaction mechanisms that share spaces with humans based on the emotional state of the human involved in the interaction. For example, a robot might assume the initial condition of the human to be neutral at the start the conversation. As the interaction continues the robot could adapt to the evolving emotional state of the user. The sensing capabilities of autonomous robots and interactive systems more generally can gather a wide range of multi-modal content, from which emotions and emotional state can be estimated.

2.2 Emotion recognition in conversation

Developing an emotionally aware robot requires solutions to a wide range of tasks from generating an emotionally-informed visual presence to integrating emotion into uttered responses, both in terms of the actual words used and the way in which they are uttered, to how the robot moves through its space, to how it presents visually. Beyond these capabilities, a key enabling technology in the development of such robots is the development of technology that understands the emotional content, the *sentiment*, of utterances made by individuals in their interaction with the robot and others. Sentiment analysis and emotion detection in conversation is key in several real-world applications, with the use of modalities such as text, speech, video

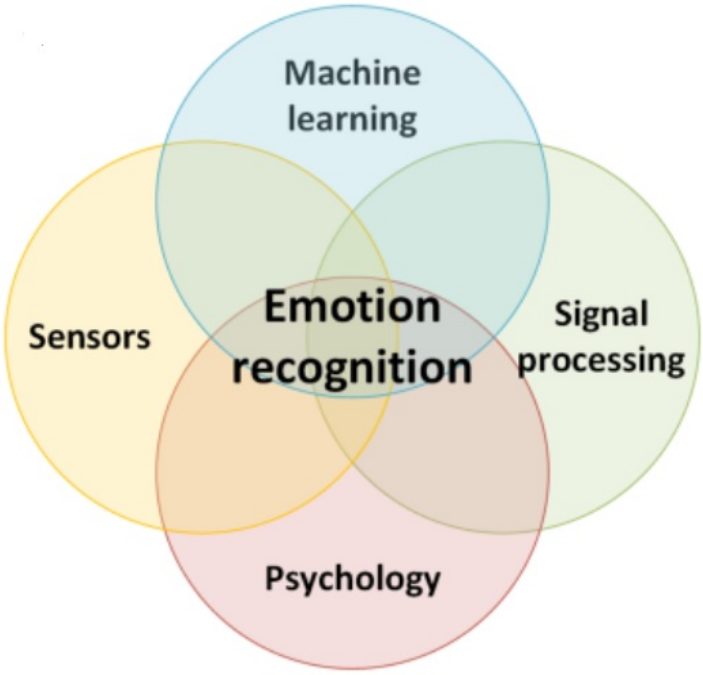


Figure 2.1: Emotion Recognition an interdisciplinary approach.

and their integration providing an understanding of the underlying emotions being presented.

More formally, Emotion Recognition in Conversations (ERC) is the task of recognizing emotions from utterances in a conversation [11]. Given a set of utterances in a conversation, as shown in Figure 2.2, along with speaker information of each utterance, the ERC task aims to identify the emotion of each utterance from a set of pre-defined emotion classes. Utterances take place in sequence and are associated with a given speaker, and the sentiment of a given utterance depends on many factors. Although there has been a long history of research in ERC, previous emotion recognition systems typically ignore conversation specific factors such as the presence of contextual information, speaker-specific information and the temporal nature of conversations and concentrate on the emotional content of the utterance in isolation [11]. However, contextual information plays an important role in ERC and this information can be used to develop effective ERC systems.

Figure 2.2: A turn-taking dialogue between two speakers A and B.

In order to understand this problem more concretely consider a dyadic (turn-taking) conversation between two speakers A and B as shown in Figure 2.2. Here U1, U3 and U5 are the utterances from speaker A and U2 and U5 are the utterances of speaker B. During the interaction the emotions change for both the speakers as emotions are the outcomes of our responses based on the situation [12]. As the interaction continues between the speakers there are various factors that need to be considered in order to understand the emotional dynamics of the conversation. Poria et al. [11] identified the following factors that affect the conversational dynamics; the topic of conversation, the speaker's personality, argumentation logic, intent, viewpoint, and the speaker's emotional state. In general, conversations can be categorized into two broad groups: task oriented conversations which are usually formal and short and non-task oriented conversation which can generally be described as "chit-chat" and may be quite long and might cover a range of unrelated topics. Still, both kinds of conversation include the above-listed factors [11].

Figure 2.3: Ekman's emotion classes categorization. Redrawn from [55].

2.2.1 Emotion categorization

The psychology literature contains many theories that deal with the conceptualization of emotion in conversations. These theories use a range of different approaches to categorize or model emotions based on distinct emotion classes or labels. These models typically assume that some set of discrete emotion categories exist. Some of the key theories are reviewed here. For a more detailed review of emotion categorization from the psychological literature, see ([52], [53], and [54]). Here we describe the two most commonly used emotion categorization theories in detail.

Perhaps the most popular emotion categorization theory is due to Ekman [55], who proposed that emotions are biological in nature and that all individuals are born with the ability to experience the same set of emotions. Within this theory he identified six basic emotions that are universally experienced in all cultures. His rationale for this is that he believed emotions that are commonly fit for the needs of life are developed

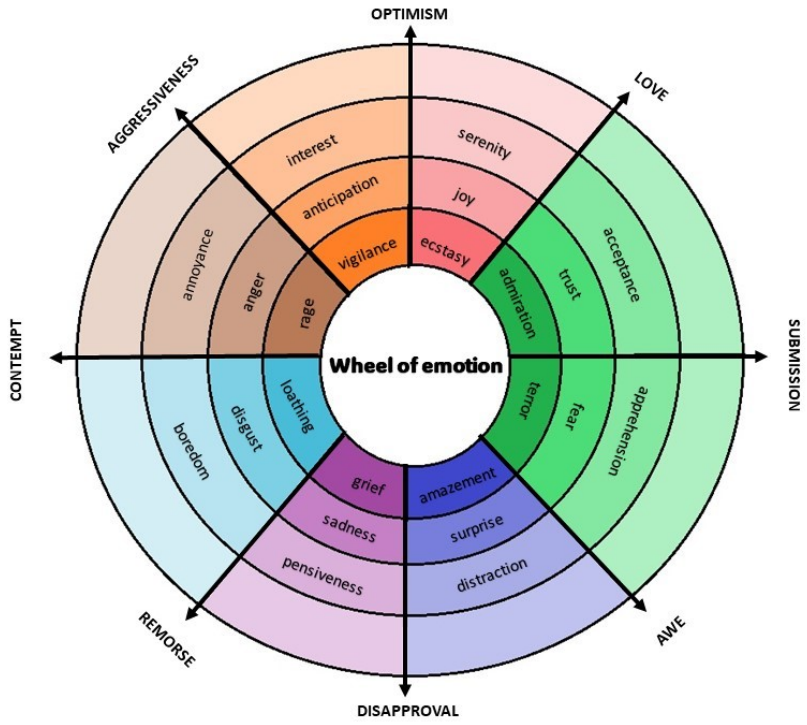


Figure 2.4: Plutchik wheel of emotions. Redrawn from [56].

early in life and are universally expressed by the same facial expressions. The emotion labels categorized by Ekman are shown in Figure 2.3 and are: *anger*, *neutral*, *fear*, *joy*, *sadness*, and *surprise*. One criticism of Ekman's labeling of emotional classes is that it does not provide a natural way of describing the relationship between them. To overcome this, Plutchik [56] visualized a wheel-shaped set of emotion categories that provides a logical way of describing feelings and the relationship between them. As shown in Figure 2.4, Plutchik categorizes emotion into eight categories with four bipolar sets: *Joy and Sadness*, *Anger and Fear*, *Trust and Disgust*, and *Surprise and Anticipation*. The reason for choosing these sets is that each primary emotion has a polar opposite. These categories are based on the physiological reaction each emotion creates and this also helps to visualize different ranges of emotion in individuals.

2.3 Text-based emotion recognition

Text-based emotion recognition has become an extremely important and well-researched topic over the last decade. Text-based emotion analysis has wide application in a large number of domains, from online transaction analysis to general understanding of textual information. Driven by these and other applications, text-based emotion analysis has been enabled by the availability of a large amount text data available through social networks, blogs, and groups where people express and convey their opinions and emotions to others.

There exists a vast literature on emotion detection from text. Text-based emotion classification typically operates at three levels; the word-level, the phrase-level and at the sentence-level. A recent survey on text-based emotion detection is given in [57]. This research surveyed the main approaches used for emotion detection in text including rule-based, hybrid, and machine-learning methods. Regardless of the computational mechanism used, the basic approach is to assign to a given text sequence one of a number of different pre-defined emotion labels. Early rule-based and statistical approaches to emotion detection (e.g., [58], [59]) have generally been replaced by machine learning models (e.g., [60], [61], [62] and [63]). The machine learning approach described in [64] is representative of a data-driven approach. Here a Long-Short Term Memory (LSTM) [65] model is used for text based emotion classification. The approach uses an emotional word vector description and a semantic word vector of the input text. An autoencoder is utilized to obtain the salient features from the emotional word vector. Features extracted by the autoencoder are concatenated with features in the semantic word vector to form the final textual features for emotion recognition for the sequence.

There exists a number of similar approaches following this strategy. For example, [66] adopts a pre-trained BERT model to classify emotions in each utterance of two common datasets, Friends [1] and EmotionLine [3]. [67] uses a BERT-based model trained on a large, manually annotated dataset using 58k Reddit comments called GoEmotions labelled for 27 emotion categories. [63] reviews transformer-based approaches for text-based emotion analysis including Generative Pre-Training (GPT) [63], Bidirectional Encoder Representations

from Transformers (BERT) [68], XLNet [69], and Robustly Optimized BERT (RoBERTa) [70].

To better understand a conversation's general emotion and user-specific emotion, [37] proposed DialogueRNN, a model based on a recurrent neural network for emotion recognition in conversation. The network extracts emotions from utterances in a conversation based on three factors; the speaker, the preceding context in the conversation, and the preceding emotion behind the conversation. To include these factors into account, the model architecture uses three gated recurrent unit (GRU) to obtain the emotion representation of an utterance for the classification. Although sequence-based neural networks have proven effective, the sequential nature of the architecture can make it difficult to capture more complex relationships between speakers in a conversation. As a consequence, graph-based approaches have become very popular for text-based emotion recognition. DialogueGCN, an advanced variant of this approach is described in [33] and the architecture of the model is shown in Figure 2.5. This framework is composed of three major modules: a sequential context encoder, a speaker-level context encoder, and an emotion classifier. To capture the contextual information from the conversations, a bidirectional gated recurrent unit (GRU) is utilized for sequential modeling of textual data to capture the contextual information present. The captured information at this stage is speaker-independent. In order to capture the speaker-dependent information from the conversations a graphical approach is used. Speaker level dependency, interdependency and self-dependency among participants are critical to SERC and ERC. Based on these relations a directed graph is constructed between the sequentially encoded utterances and interactions between the speakers. A feature transformation process transforms the contextually encoded features into more enriched speaker-level context features. This utilizes the local information from the neighbors. A third module concatenates the contextually encoded feature vectors (from the sequential context encoder) and the speaker-dependent context features (from the speaker-level context encoder) to form the final utterance representation. This representation is then classified using a fully connected network into different emotion categories for the appropriate datasets. This work relies on IEMOCAP [2], AVEC [71], and MELD [1] datasets. Although all of three datasets are multimodal, this work only utilizes the textual information to perform emotion recognition in conversation. Emotion is also

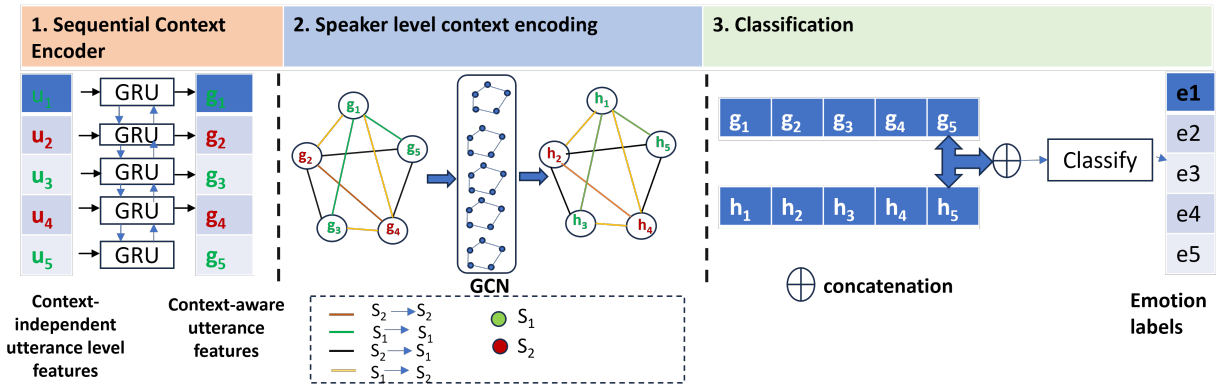


Figure 2.5: DialogueGCN model architecture. Redrawn from [33].

influenced by other factors. [4] proposes a method known as COSMIC for emotion detection in conversations using different aspects of common sense including events, mental states and casual relations to build the model and learn from it. The model learns the interactions between speakers. During a conversation, there are various factors including the topic of the conversation, speaker's personality, logic, viewpoint, and intent which play major role in effecting the emotional dynamics of the speaker. Along with it, the utterances are also associated with the mental state of the speaker. For example, if any recent event has happened with the speaker that has affected their emotional state or the prior emotional state when it is uttered. This model uses this commonsense knowledge such as the reason for an event or situation that can explain the emotional shift in the conversation. Mainly, this model tries to address the challenges in emotion recognition task including detection of emotional shift in the utterances within a conversation which normally is omitted by the models in the literature.

2.4 Speech emotion recognition (SER)

Text is not the only form of language that has been used for building emotion recognition systems. Speech is considered to be the most natural form of human expression and so it is only natural that speech based emotion recognition systems are very popular. Detecting the embedded emotion expressed in spoken ut-

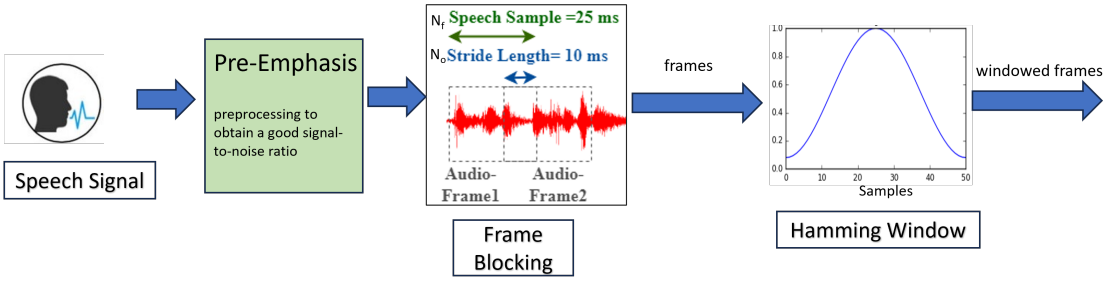


Figure 2.6: Speech Signal Framing and Windowing. Redrawn from [25].

terances encoded through the acoustic signal is known as Speech Emotion Recognition (SER). Determining the emotional state of humans through their vocalized speech can help to facilitate natural interactions with chatbots/robots using voice either in isolation or in part of a multimodal approach.

2.4.1 Traditional techniques for SER

Emotion recognition systems based on digitized speech are typically comprised of three fundamental components; signal preprocessing, feature extraction, and classification [72]. The feature extraction process extracts distinctive acoustic features that characterize the utterance which directly influence the classifier performance and impact the overall efficiency and computational time of the recognition system [25]. Processing the signal normally involves signal pre-processing which includes pre-emphasis, denoising and segmentation steps to capture important characteristic information of the signal [73] without the noise.

Figure 2.6 illustrates a typical preprocessing step in detail. Audio signals are first captured into frames. The frame blocking or framing is a preprocessing step that divides the original speech signal into small blocks or frames with N_f being the speech sample length and N_o is the overlap or stride length. Overlapping the frames is important as it helps to prevent information loss between the adjacent frames. A speech signal is a non-stationary signal and thus the statistical properties of the signal changes over time. During framing it is assumed that the statistical properties of the signal are constant within the frames [74]. Characteristic properties of the signal are extracted from these small audio frames. These extracted feature vectors should

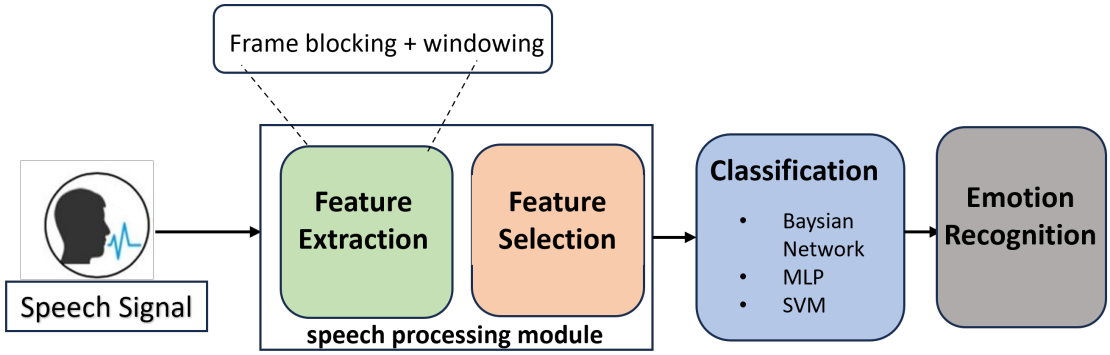


Figure 2.7: Classical speech emotion recognition approach. Redrawn from [25].

have a compact form and should capture the important signal information. Finally, extracted feature vectors are mapped to relevant emotions of interest. Figure 2.7 depicts a simplified speech-based emotion recognition system building on this frame-based representation.

In the first stage of speech-based signal processing, preprocessing attenuates the noisy components. The second stage involves two modules, feature extraction, and feature selection. The required features are extracted from the processed speech signal and then feature selection is performed. Examples of widely used acoustic features include mel-frequency cepstral coefficients (MFCCs), linear prediction cepstral coefficients (LPCC), short-time energy, and the fundamental frequency (F0) [12]. During the third stage, classifiers are utilized to characterize these features. Traditional classifiers for emotion recognition include Bayesian Networks (BN) [75], Maximum Likelihood Principle (MLP) [76] based approaches, and Support Vector Machines (SVM) [71]. In addition to these and other linear classifiers, there exist non-linear classifiers which are used when the signal is considered to be non-stationary. Many non-linear classifiers have been used for SER, including Gaussian Mixture Models (GMM) [77] and Hidden Markov Models (HMM) [78].

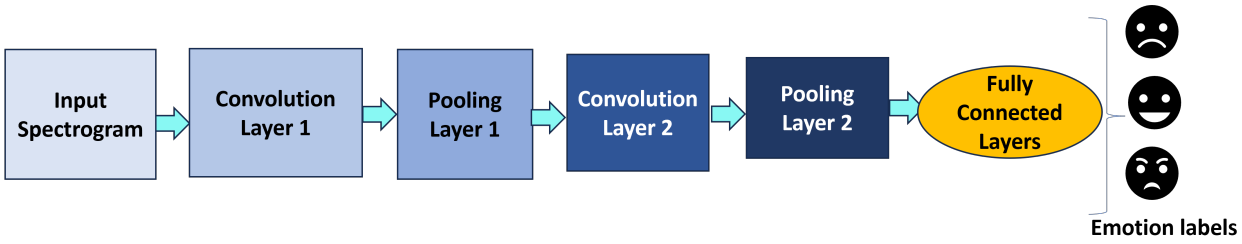


Figure 2.8: Detail overview of a CNN-based architecture. Redrawn from [25].

2.4.2 Deep learning methods for SER

As traditional techniques such as Gaussian Mixture Models (GMMs) for acoustic modeling in speech recognition system are displaced by Neural Networks, many researchers (e.g., [79]) have employed Deep Neural Network models for SER. Deep learning methods have found applications in a wide range of tasks including natural language processing (e.g., [80]), pattern recognition (e.g., [81]), image recognition (e.g, [82]), and in speech emotion recognition (e.g., [83, 84, 85]). The layers in deep learning architectures provide the network the ability to automatically extract features from signals including raw speech data and to learn data patterns that enhance classification performance. Data-driven approaches have generally replaced the process of careful feature extraction and selection. Deep learning architectures including Convolutional Neural Networks (CNNs) (e.g., [25]), Recurrent Neural Networks (RNNs) (e.g., [86]), Long short-term memory (LSTM) (e.g.,[65]), and autoencoders (e.g., [87]) are widely used and have provided improved speech emotion recognition systems accuracy. The detailed architecture of a CNN model for speech emotion recognition is shown in Figure 2.8.

LSTM's are neural networks that integrate information over time and are capable of processing time series data such as speech [88]. [79] investigated how LSTM recurrent neural networks can be utilized for SER. In [89] a dual-sequence LSTM (DSLSTM) architecture is proposed for sppech emotion recognition. The model takes MFCC features and mel-spectrogram produced from audio signal into account to predict the emotion label of the input audio utterance. A classic LSTM processes the MFCC features and the proposed DSLSTM

take the features from the two mel-specrograms. Finally, the outputs from both the LSTMs are averaged to perform the classification. In order to capture and learn local and global emotion-related features from speech [90] proposed to use a 1D CNN LSTM network to recognize speech emotion from audio clips and a 2D CNN LSTM network to learn global contextual information from the handcrafted features.

[88] proposed an attention-based dense connection LSTM model which adds weight coefficients in each dense layer to distinguish differences in the emotional information between layers to reduce the interference of redundant information from the bottom layer with the information from the top layer. They tested their methodology on the eNTERFACE and IEMOCAP datasets and showed improved performance as compared to the baseline methods available at that time.

For SER, a CNN is often the first architectural choice to combine with other discriminative or generative methods. Many early approaches, such as [91] used a CNN to learn effective salient features for speech emotion recognition. CNN methods based on classic networks such as AlexNet [92], CaffeNet [93] and GoogLeNet [94] have been used to learn emotional patterns from the spectrogram associated with an utterance and predict the corresponding emotions. These models are commonly used for transfer learning, see [95], [96] for example. The approach described in [96] is representative. [97] proposed a suggests a convolutional neural network (CNN) based model with rectangular shape filters to extract the deep frequency emotional features from the spectrograms generated from the audio recordings. The model is composed of three blocks with several convolutional layers followed by a pooling layer to extract the features with additional dense layers and a softmax layer as output. This method of extracting deep emotional features from spectrograms proves to be an efficient technique as compared to other deep learning baselines.

In autoencoders, speech data is used to initialize the input layer. An encoder layer is then used to extract hidden features from the input speech data. A hidden layer is then used to represent a low-dimensional set of hidden features. Using coupled encoder and decoder units, deep autoencoders have been used to extract low-dimensional features from utterances and have proven to be particularly effective in emotion recognition.

For example, [87] proposes a classifier with a deep autoencoder based on a multilayer perceptron. They utilize the capability of the autoencoder by initializing the weights corresponding to the links between the input layer, the hidden layer and the output layer during a pre-training phase.

In the literature there also exist hybrid methods for speech emotion recognition. [98] combines a convolutional neural network with a Bidirectional LSTM that captures long range temporal dependencies. [99] compares the results of their model based on CNN and LSTMs to the results achieved with conventional classifiers. [100] applies a deep neural network composed of convolutional and LSTM layers for emotion recognition. Although they can be effective, the main issue with deep neural network models is that they require a large amount of data during training to obtain competitive performance.

2.5 Graph neural networks (GNNs)

Although, conventional deep learning techniques have achieved considerable success in modelling sequential data (text data) and images (with CNN) there still exist problems when the relationships in the data are not well represented as 2D data (spectogram) or linear structures. For more complex conversations, relationships between the data can often be better represented in the form of a graph. This type of representation has proven effective in other domains. For example, in e-commerce, a graph-based learning system can leverage interactions between users and the kind of products they use to make highly accurate recommendations. In chemistry, molecules can be modeled as graphs, and their bioactivity for drug discovery modelled in terms of this graph as shown in Figure 2.9. In a citation network, articles or research papers can be linked to each other via citationships, and categorizing them into different groups can be performed using this representation. Interactions in a social network can be modeled as a graph as shown in Figure 2.10. In this interaction graph the nodes are the users and the edges show their ties with each other. This graph can be used in a node classification task to make suggestions to the users for new connections or friends to connect with that share the same mutual friends. The GNN can also be used to design a recommendation system to

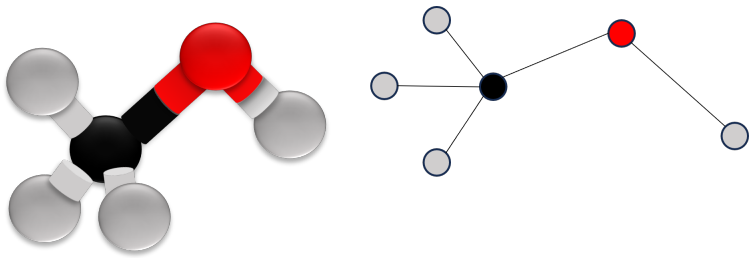


Figure 2.9: Molecules modeled as graphs redrawn from [101]. The image on the left describes the 3D molecular structure while the image on the right shows its representation as a graph. In the graph structure, the nodes are the atoms and the edges represent the bonds between them.

suggest products, music or movies based on common interests.

Graph Neural Network (GNNs) [102] are deep learning models aiming at addressing graph-related tasks in an end-to-end manner. Many GNNs are used explicitly to extract high-level representations. In the past the complexity of graph data has presented significant challenges for machine learning algorithms. As graphs can be irregular, a graph may have a variable size, and nodes from a graph may have different numbers of neighbors, resulting in some important operations (e.g., convolutions) that are easy to compute in the image domain being difficult or computationally expensive to apply in the graph domain. Furthermore, a core assumption of many existing machine learning algorithms is that nodes are independent of each other [103]. This assumption does not hold for graph data in general as each node is related to others by links of various types, such as citations of the articles or research papers, friendships that occur in a social network, and other interactions [103]. In order to help address this, [103] proposes a taxonomy that divides GNNs into four categories; recurrent GNNs (RecGNNs), convolutional GNNs, graph autoencoders, and spatial–temporal GNNs.

- Recurrent GNNs (RecGNNs): Most early GNN's fall in the category of recurrent GNNs. The architecture aims to learn representations using recurrent neural architectures. For information to propagate in RecGNN same weights matrices are applied iteratively until a stable equilibrium is reached.

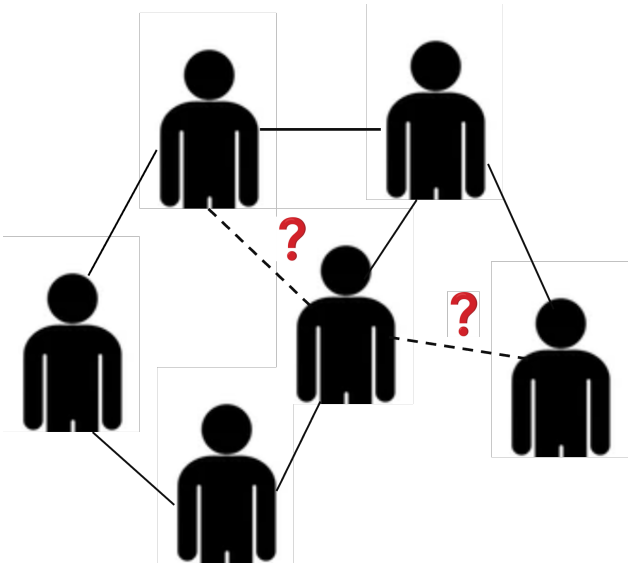


Figure 2.10: Social network interaction graph. Redrawn from [101]. In the graph structure, the nodes are the users and the solid edges represents connections or friendships between the users.

- Convolutional GNNs (Conv-GNN): These GNNs generalize the operation of convolution from grid data to graph data. The main idea here is to generate a node representation by aggregating its own features with the features of the node's neighbors [101]. The main difference in operation between RecGNNs and Conv-GNN is the propagation of information. In contrast to RecGNNs, convolution GNN applies different weights at each iteration.

- Graph Autoencoders: These GNNs are unsupervised learning frameworks comprised of an encoder and decoder. The encoder module encodes nodes/graphs into a low-dimensional space. It is considered to be a powerful graph embedding technique that preserves the graph structure. With the decoder, it can reconstruct the original graph data from the encoder output information. GAEs are used specifically to learn network embeddings and help to build efficient systems for various tasks including link prediction and node clustering.

- Spatial-Temporal GNNs: These GNNs learn hidden patterns from spatial-temporal graphs, which consider spatial dependency and temporal dependency at the same time. With these features, these

GNNs are adopted for different variety of applications such as traffic speed forecasting [104]. To perform this task for forecasting a specific road traffic, the road condition will indeed depend on its adjacent road conditions. The spatial-temporal GNN will consider spatial dependency and can be helpful for this task.

Figure 2.11 presents the general pipeline design of a graph neural network. The first step in the pipeline is to identify a graph structure or architecture relevant to the application. The next step is to specify the graph type that best represents the problem being solved. In the next step, a loss function is chosen and finally, the GNN model is designed using the computational modules. Generally, graphs are categorized as follows:

- Directed/Undirected Graphs: Edges in directed graphs are directed from one node to another while the edges in an undirected graph represent a two-way relationship where each edge can traverse in both directions and can be regarded as two directed edges.
- Homogeneous/Heterogeneous Graphs: Nodes and edges in homogeneous graphs have the same types, while nodes and edges may have different types in heterogeneous graphs.
- Static/Dynamic Graphs: The dynamic graphs represent the graph data with nodes representing an entity and the edges show the relationship or interactions between the nodes. These relations or interactions vary with time the graph is regarded as a dynamic graph. Otherwise, it is a static graph.

2.5.1 GNN framework tasks

Once the graph structure is determined then the output of the GNNs can focus on different graph tasks. The following mechanisms can be identified [101]:

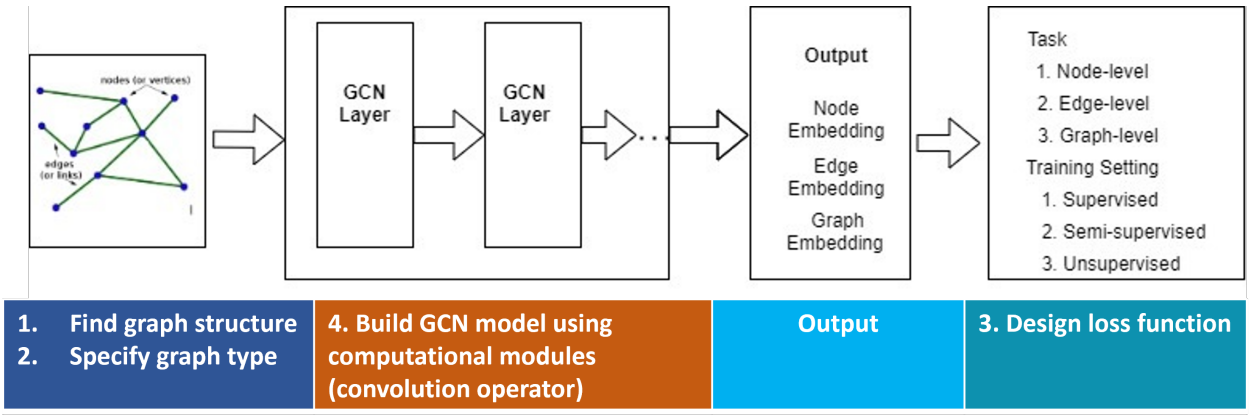


Figure 2.11: General design pipeline for a GNN model. Redrawn from [101].

- Node Level analysis: In this analysis the output is related to node regression and node classification tasks. Examples include RecGNNs [105] and ConvGNNs [106]. This approach extracts high-level node representations using graph convolution. GNNs can perform node-level tasks in an end-to-end manner with multi-layers architecture and a Softmax layer as the output layer. For example, node classification can be used to classify scientific papers in a citation graph where labels are only available for a small subset of nodes, and the GCN must predict the correct label for the older node.

- Edge Level: Outputs related to edge classification and link prediction tasks are considered under this framework. Using the two nodes associated with the given edge and nodes' hidden representations obtained from GNNs as inputs, a similarity function is utilized to predict the label or connection strength of an edge [107]. For example, link prediction predicts whether two nodes in a network are likely to have a link such as a friend recommendation in social network [108] and movie recommendation [109].

- Graph Level: Outputs related to graph classification tasks are considered in this framework. For example, graph classification task are used in bio-informatics applications to predict the function of a protein structure, predicting if the cells are cancerous or not, predicting if a protein is enzyme or not, etc., as graphs can be used to model complex relations. Similarly, another task can be performed including

graph matching and graph regression which require the model to learn the graph representations.

GNN's can also be categorized by the way in which they are trained:

- Supervised setting: This task provides labeled data for training.
- Semi-supervised: This task gives a small amount of labeled nodes and a large amount of unlabeled nodes for training.
- Unsupervised setting: The GNN is presented with unlabeled data for the model to find patterns. For example, Node clustering is a typical unsupervised learning task [110].

To build a GCN model different computational modules are used. Commonly used modules include:

- Propogation module: This module propogates information between the nodes in order to aggregate the feature and logical information. Convolutional operators and recurrent operators are used to aggregate information from the neighbouring nodes in the model.
- Sampling module: This module is generally used when the size of graphs is large and is usually combined with the propagation module.
- Pooling module: This module is needed to obtain high level graph representations and to extract information from the nodes.

2.5.2 Spectral graph convolution

The convolutional operator used in the GNN model generalizes convolutions in other domains to the graph domain. Following [31], this work adopts the spectral GCN [111] graph architecture to extract the utterance level features. Spectral convolution provides a mathematical framework to design filters. The eigenvectors

and eigenvalues of the graph Laplacian matrix facilitates the processing and analysis of signals on graphs. As per the theory of graph signal processing [112], graph convolution is given by

$$h = x_i * w \tag{2.1}$$

where h is the convolution output, x_i is the input vertex feature vector, * is the convolutional operator, and w is a learnable graph convolution kernel.

A spectral graph convolution is defined in the Fourier domain by applying the filter on the input signal x_i. Equation 3.2 is equivalent to the product of x_i and w_u as

$$\hat{h} = \hat{x_i} \otimes \hat{w_u} \tag{2.2}$$

where $\hat{h}$, $\hat{x_i}$ and $\hat{w_u}$ denote the output, input features and the convolutional filter in the graph spectral domain. The cyclic graph structure as adopted in [31] is important because of the special structure of the graph Laplacians, which significantly simplifies the computation and results in a lightweight spectral GCN operations as shown below.

The exact graph convolution is given by $\hat{H} = (U^T X)(U^T W_u)$; then $H = U\hat{H}$, where U is the eigenvector. For computation in matrix form, for the GCN k^{th} layer, the graph convolution propagation is evaluated as:

$$H^{(k+1)} = U\Big((U^T H^{(k)})(U^T W_u^{(k)})\Big) \tag{2.3}$$

Following [113], a multi-layer perceptron (MLP) can be used to learn the convolution kernal (W_u). Adopting the approach described in [31] the convolution kernal as described above can be learned using MLP as

$$H^{(k+1)} = U\Big(MLP(U^T H^{(k)})\Big) \tag{2.4}$$

where, only the MLP parameters are learned which makes this approach more scalable and efficient.

2.5.3 Graph pooling modules

After a GNN generates node features, we can use the features to formulate a given task. It is computationally challenging to use all of the available features directly. In order to improve the computational capability

of the problem, a downsampling strategy is typically used. This strategy is known as graph pooling. The pooling operation aims to reduce the number of parameters by downsampling the nodes to generate a more compact representation and to avoid overfitting.

In order to obtain a compact representation of the graph level, GNNs are often combined with pooling layers along with Softmax layers. Based on the task objective and the graph network requirement, different pooling layer operations can be used including mean, max and sum pooling [114]. Calculating the mean, max or sum value in the pooling window is computationally efficient.

- Max pooling: Pools a graph by computing the maximum of its node features.
- Sum pooling: Pools a graph by computing the sum of its node features.
- Min pooling: Pools a graph by computing the minimum of its node features.

Once trained on a corpus of data, a GNN can be applied to a novel input to obtain a feature representation either at the edge, node or graph level.

2.5.4 SER using graph convolution

A graph neural network (GNN) structure allows for end-to-end learning of prediction pipelines whose inputs are graphs of arbitrary size and shape and which generates embedding vectors of nodes based on the properties of their neighborhoods. GNN's are widely used in dialogue systems to model conversations and model relationships between different modalities as they have the capability to capture the rich relationships and interdependancy between data. For example, DialogueRNN [37] uses a recurrent neural network to extract emotions based on three factors: the speaker, the preceding context in the conversation, and the preceding emotion in the conversation. DialogueGCN [33], is a more advanced variant of this approach. This framework is based on a graph convolutional network that includes the speaker level, sequential context encoder, and an

emotion classifier. For speech emotion recognition, there also exist a number of graph-based approaches. [31] proposes a deep graph based approach for speech emotion reocgnition. The authors model the speech signal as a cycle graph or a line graph and cast the emotion recognition problem as a graph classification problem. Tranforming the speech utterances into graphs resulted in an efficient, compact and lighweight model for SER. [22] proposes another graph classification approach for SER. The authors highlight the importance of transforming speech signals into a graph structure and addressing the problem of variable length speech inputs. Because clipping and padding the utterances should be avoided for variable length sequences as it can result in the loss of emotional information in speech utterances, they used GraphSAGE [32] which is a general framework model for inductive embedding. The GraphSAGE model uses node features such as node degree and node profile information to learn an embedding function that generalizes to unseen nodes as compared to approaches that use matrix factorization for learning embedding. Following this, at the next stage multi-head attention pooling (MHAPool) [115] is used to obtain the graph-level emotion vector representation.

2.5.5 Multi-modal emotion analysis

Multimodal approaches combining textual, audio, video and other signals have resulted in highly effective emotion recognition systems. Multimodal ERC is influenced by the fact that humans display emotions through a range of different cues and algorithm can use this information to perform emotion recognition. A hybrid fusion process, referred to as a multimodal attention network (MMAN) [116] includes audio, visual and textual signals in speech emotion recognition. This work proposes a multimodal focus mechanism, cLSTM-MMA, which promotes attention across three modalities and fuses information selectively. [117] proposed an approach for multimodal emotion recognition. This mechanism uses a combination of cross-modal attention along with a raw-waveform encoder-based convolutional neural network. To extract the features from raw audio signals they use an audio encoder using a one-dimensional convolutional model. A text encoder is used to extract semantic features from textual data. A cross-modal attention mechanism

is used between the audio and text features to obtain enhanced performance in emotion recognition task. [118] describes a multimodal deep learning model that utilizes facial images of the TV actors annotated with the textual description of the action played by the actor. The reported results showed the importance of incorporating multi-modality to achieve good performance in face recognition system.

Graph based approaches have achieved state of the art results in the multi-modal domain as GNNs can easily capture long-distance contextual information in ERC tasks through relational modeling [119]. ConGCN [120] models represent utterances and speakers as nodes, context dependencies, and speaker dependencies as edges, for multi-participant conversational emotion detection. ConGCN takes into account textual features and acoustic features. MMGCN [121] considers both multimodal information and long-distance contextual information through a graph-based model. [122] utilizes the capability of heterogeneous graph neural networks to capture information from multi-dimensional sources including dialogue history, its emotion flow, facial expressions, audio, and the speakers' personalities. Their model is constructed using a heterogeneous graph-based encoder and an emotion-personality-aware decoder. [119] proposes a novel graph-based multimodal model GraphMFT that captures intramodal contextual information and inter-modal complementary information by modelling a conversation as three parallel graphs: Video-Audio, Video-Text, and Audio-Text where each graph contains information related to only two modalities.

2.6 Acoustic speech features

A critical challenge in SER problems is determining the features that influence emotion recognition in the raw speech signal [123]. The existence of different contexts, genders, speakers, and speaking styles makes it difficult to identify the best acoustic feature set because these properties have direct affect on the features such as pitch, and energy [124]. Some of the acoustic speech features reported in the literature are given in Figure 2.12 and are described below. These features are divided into qualitative, spectral, continuous, and Teager energy operator (TEO-based) features [12, 125]. Commonly used speech acoustic features include

Melfrequency cepstral coefficients (MFCCs), zero-crossing rate, voice probability, the fundamental frequency (F0), and frame energy.

- Energy and pitch: Energy and pitch are the primary continuous acoustical features used in emotion recognition. Arousal refers to the amount of energy required to express a certain emotion. According to [124], the arousal state of the speaker affects the overall energy, energy distribution across the frequency spectrum, and the frequency and duration of pauses in the speech signal.
- Fundamental frequency: The fundamental frequency of the pitch signal, also known as the glottal waveform, because of its dependency on the subglottal air pressure and tension of the vocal folds which carries emotional information. The source of the pitch signal is the vibration of the vocal folds. The time elapsed between two successive vocal fold openings is called the pitch period T, while the vibration rate of the vocal folds is known as the fundamental frequency of the phonation F0, or pitch frequency [123]. For example, music like speech with surprise or joy emotion is associated with a high glottal volume velocity and emotions like anger or disgust is associated with low volume velocity.
- MFCC: In addition to time-dependent continuous features such as pitch and energy, spectral features are often selected as a short-duration speech signal representation. In order to comply with the spectral distribution of the auditory system, the estimated spectrum is often passed through bandpass filters or critical bands. MFCC is a frequently used spectral feature which leverages the human auditory frequency response with the help of Mel-Scale frequency response.
- TEO-based features: TEO-based features are additional excitation signals such as harmonics and cross harmonics [12].

Appropriate and efficient feature extraction from speech data is important as it helps to reduce the computational complexity and improves recognition accuracy. Features can be extracted manually or automatically. Speech signals pre-processing is a significant step before extracting features in the development of robust and efficient SER systems.

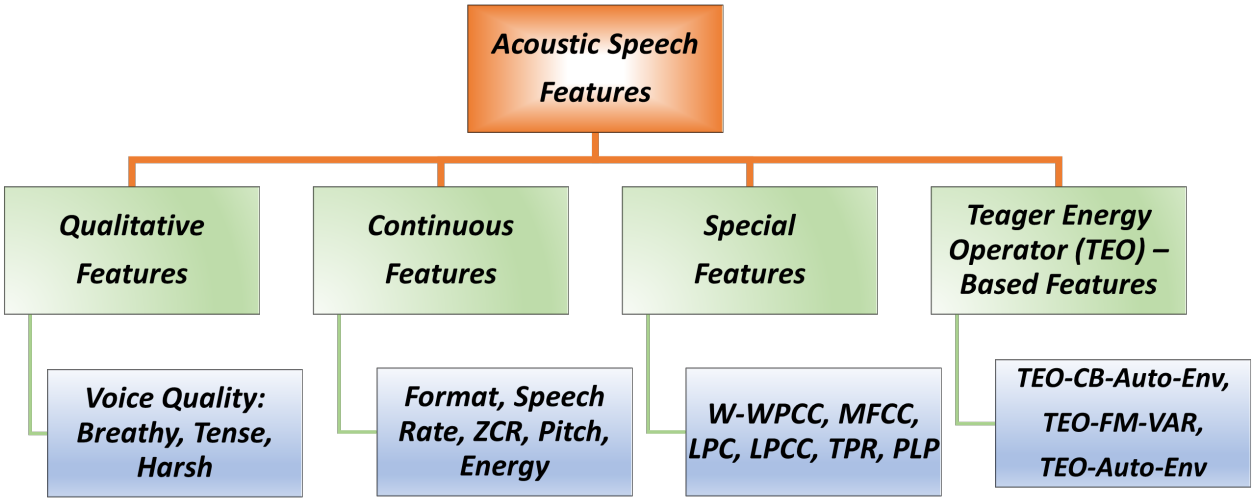


Figure 2.12: Different categories of speech acoustic features. Redrawn from [12].

Manually extracted features are domain-dependent. There exist several automatic feature extractor toolkits for SER including OpenEAR[127], Kaldi [128], COVAREP v1.4.1 [129], pyAudioAnalysis [130], Python Speech Features [131] and OpenSMILE v3.0 [126]. Table 2.1 describes the commonly extracted audio features from these toolkits. This work uses the OpenSMILE toolkit. It is chosen due to the reference work [31] which also uses the same tool to extract speech signal features. Details related to OpneSMILE are given below.

2.6.1 OpenSMILE 3.0

The Munich open-Source Media Interpretation by Large feature-space Extraction (OpenSMILE) toolkit [126] is a modular and flexible feature extractor for signal processing and machine learning applications. OpenSMILE can work for real-time online processing and can also be used to work off-line in batch mode for the processing of large datasets. OpenSMILE extracts features incrementally as the data arrives. This is a

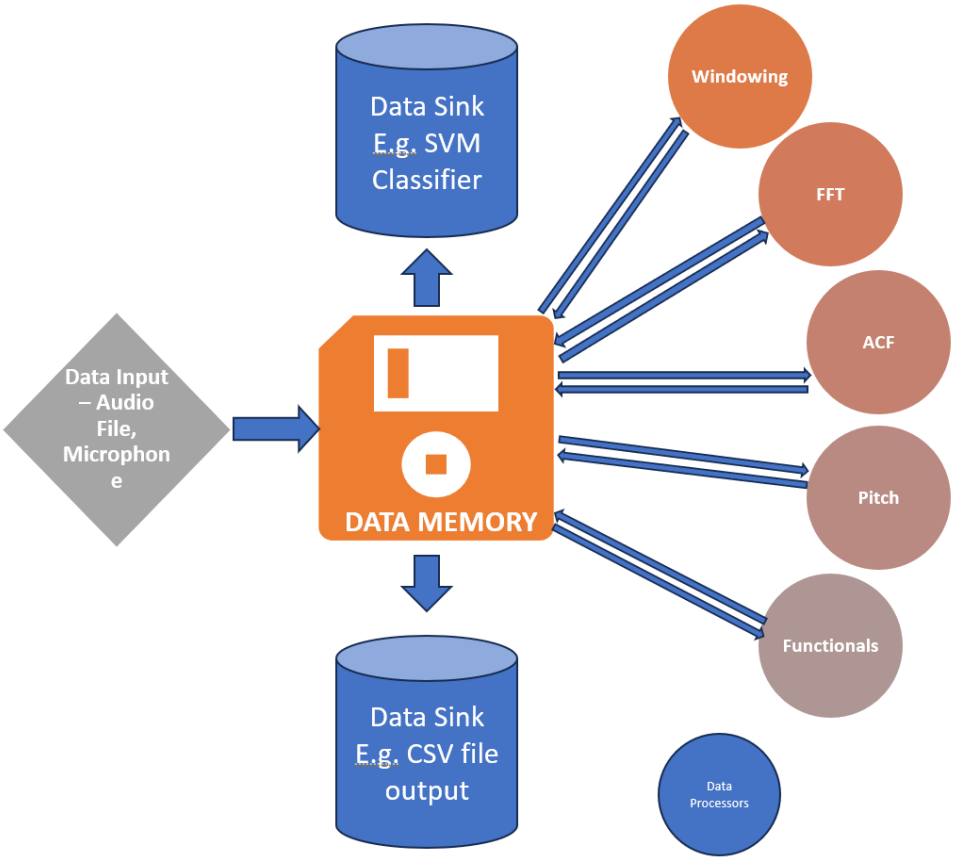


Figure 2.13: Description of OpenSMILE Architecture. Redrawn from [126].

feature rarely found in other tool kits for speech recognition feature extraction. Figure 2.13 shows the overall data-flow architecture of OpenSMILE. Within OpenSMILE the Data Memory act as a central link between all Data Sources. OpenSMILE writes data from external sources to the data memory. OpenSMILE is capable of extracting Low-Level Descriptors (LLD's) and applying various filters, functionals, and transformations to them. The LLDs that are commonly extracted using this tool are listed in Table 2.1.

2.7 Baselines methods for comprehensive analysis

Baselines for utterance-only models and conversational SER models are listed below. These models use the IEMOCAP dataset for evaluation.

Table 2.1: Common Low Level Descriptors(LLD's) extracted in the OpenSMILE toolkit [132].

Feature	**Description**
Speech-related features	Signal energy, voice quality (shimmer and jitter), Mel Frequency Cepstral Coefficient (MFCC), Pitch, Loudness, Mel/Bark spectrum bands, formants, LPC, spectral shape descriptors
Music-related features	CHROMA, CENS features, Pitch classes (semitone spectrum), Weighted differential
Statistical feature	Means, Extremes, segments, peaks, samples, percentiles, zero-crossing, Linear and quadratic regression, Onsets, DCT coefficients

2.7.1 Utterance only SER Models

Many studies use utterances to extract audio features for emotion recognition task. Speech emotion recognition at the utterance level can be formulated as a many-to-one sequence-to-sequence learning, where the input sequence is the stream of acoustic frames and the output sequence is the emotion label [36]. [36] applied the attention mechanism on the Bidirectional (BLSTM) model for SER. As reported in the paper, they found that using an attention mechanism resulted in an improved selection of frames being achieved which resulted in an improved weighted accuracy. Another work [133] adopted an emotion-pair framework that is able to create a differential feature space for every emotion-pair (two different emotions) to generate more precise emotion bi-classification results. The bi-classifier used in their work is the Bayesian Logistic Regression (BLR) classifier which is trained for a specific emotion-pair and to distinguish the emotions in that emotion-pair. [134] uses a deep recurrent neural network (RNN) with an attention mechanism, to learn

the emotionally relevant short-time acoustic features and the temporal aggregation of those features into a compact utterance-level representation. Results reported with the IEMOCAP dataset shows that learned features with this model provide a higher accuracy. In order to capture richer emotional information from speech [135] proposed a Convolutional Recurrent Neural Network (CRNN) to extract high-level statistic functionals (HSFs). The HSFs are the global dynamic version of LLDs and are calculated as statistics of LLDs, such as their mean, maximum, minimum, variance, kurtosis, and skewness. Thus, to fully extract the emotional information in speech signals the authors utlize HSFs and a CRNN model that learns the features from the spectogram and proposed HSF-CRNN, a two channel SER system. The system is evaluated on the IEMOCAP corpus and achieved a weighted accuracy of 60.35% and unweighted accuracy of 63.98%. Another work [136] proposed a model with a multilayer CNN network stacked on a LSTM. The results reported in the paper shows that the proposed construct of the classification block, where the convolutional layer captures high-level abstraction, the LSTM layer performs long-term temporal modelling, and the fully connected layer performs discriminative representations, offers improvements in the performance of emotion recognition.

There exist graph-based models that use utterances for the emotion recognition task as well. A straight forward GCN [137] is the most common baseline for graph based models. But there are other approaches including spectral GCN [138] which is specially designed for the node classification task. Another work described in [139] developed a framework for learning graph representations in arbitrary graphs. The proposed approach is known as PATCHY-SAN. The main objective of the framework is to solve two problems: (i) to identify the node sequences for which neighborhood graphs are created. These neighborhoods are the receptive field of a CNN network that allows the framework to learn deep and complex graph representations and (ii) compute a normalization of graphs which requires a mapping of the neighborhood graphs representation into a vector space. PATCHY-SAN is an efficient approach and outperforms graph kernels in terms of accuracy and computation speed for graph classification problems. [140] introduces DIFF-POOL a differentiable graph pooling module that can be adapted to various graph neural network architectures

in a hierarchical and end-to-end fashion. This framework adds a differential assignment at each layer of a deep GNN, and maps the nodes to a set of clusters by learning a soft cluster assignment at each layer of a deep GNN. Then the output is sent to the next GCN layer. This methodology helps to generate deep GNNs by stacking GNN layers in a hierarchical fashion. [31] uses the concept from the above two graph baselines models and proposes a GCN-based methodology for SER that outperforms the deep learning models and the graph based models. In this compact and efficient GCN model the speech utterances are transformed into graphs which makes the convolution process simple and thus achieves a higher performance compared to existing models. The GCN architecture assigns an emotion label to each speech sample transformed into graphs. This approach forms the underlying model used in our context-free stage.

Recent work described in [22] on the SER problem transformed variable length utterances into graphs for improved retention of the emotional information in utterances as cutting and padding the utterance can lead to emotional information loss. After the feature extraction graph is constructed and node embedding learning is performed using GCN models including standard GCN [141], GAT [142] and, GraphSAGE [143] which is an inductive graph convolutional network that can generate node embeddings for unseen data. The next step is to apply pooling function in order to obtain graph-level representation. Different pooling functions are explored in [22]. Results showed that a combination of GraphSAGE, which is used for node/frame embedding learning, combined with and multi-head attention pooling (MHAPool) [144] which is employed to obtain graph/utterance representation, received higher accuracy when evaluated on the IEMOCAP dataset. These models do not consider speaker dependency during modeling the utterances for the recognition task.

2.7.2 Conversational SER Models for IEMOCAP

Instead of considering only isolated utterances for emotion recognition, there is evidence that considering context-sensitive and speaker dependency can plays an important role in emotion recognition system. [23] proposes several approaches for conversational emotion recognition (CER) including the transformer model,

RESNET-34 and BiLSTM. The authors explored the effect of context by comparing the models trained on single utterances and conversations. In order to leverage both local and global context information, the authors use RESNET-34 for modeling local context in each layer and globally by stacking the layers. This research also explores together joining the RESNET model for its strength in modeling the context information locally and globally and a transformer architecture. This combination improved the accuracy of the emotion recognition task. The IEMOCAP dataset is evaluated by the proposed method and the results showed that the combination of RESNET with transformers proved to have a high F1 score indicating the importance of including context information for the emotion recognition task. Based on the results, our approach takes into account conversational contextual information, temporal information, and speaker dependency in a graph convolutional network-based approach. Incorporating a graph-based network methodology along with the important conversational factors should improve the performance of the system for speech emotion recognition in conversations (SERC)

2.8 Conversational Models for the multiparty MELD dataset

Models that use the MELD dataset for evaluation are described below.

2.8.1 Deep learning models

Bidirectional Contextual LSTM (BC-LSTM) [145] performs context-dependent fusion of multisequence data. This approach only considers the contextual information in a conversation. Conversational Memory Network (CMN) [35], models separate contexts for both self-speaker and other-speaker in an utterance for emotion detection in two-speaker conversations, but ignores the global contextual information of the utterance. Interactive Conversational Memory Network (ICON) [146] , models separate contexts for both self-speaker and other-speaker to an utterance for emotion detection in two-speaker conversations, and simultaneously

incorporates the global contextual information of the test utterance. Another work, DialogueRNN [37], employs three GRUs to model the speaker, the context from the preceding utterances, and the emotion of the preceding utterances. This approach considers different speakers in a two-speaker scenario.

2.8.2 Graph based models

[120] proposed a graph-based conversational convolutional neural network model (ConGCN), to model both context-sensitive and speaker-sensitive dependence for emotion detection. From the conversation dataset, each utterance is represented as a node in a graph, with an edge between the two utterances in the same conversation to symbolize contextual dependence. On the other hand, to model the speaker dependency, each speaker of the whole corpus is represented as a node and an undirected edge is constructed between each utterance and its speaker. Thus, this methodology is a graph-based approach and considers the task of emotion recognition on utterances as a node classification problem. The model is evaluated on the MELD dataset and has achieved good accuracy.

Recently, [34] proposed a graph-based neural network based approach to model context sensitive and speaker sensitive dependencies for emotion recognition in conversation. They use acoustic, textual and multimodality for the evaluation of the proposed methodology. This method leverages context and speaker dependencies by modeling the conversation using a directed graph. For the utterance level feature extraction process a self attention mechanism is used to extract the features from the frames. For speaker level context encoding a GCN architecture is used. A directed graph is constructed to model the conversation in which there are two types of nodes; utterance nodes and speaker nodes. To model the context sensitive dependency, an edge is constructed between the utterance nodes from the same conversation. They also implemented a relation reduction technique where each utterance node has edges connecting the nodes with the immediate utterance of the past, and the immediate utterance of the future. To model the speaker, edges are constructed between the utterance node and the speaker node, along with their relative positions in the conversation. In

this way the conversation graph is constructed. The model is evaluated on the multiparty MELD dataset and achieved improved accuracy compared to baseline models.

2.9 Speech corpus

Many sentiment-related datasets are constructed using artificial data e.g., actors performing the utterances. This raises a critical question about potential biases or differences between real-world emotions and the acted ones. When considering speech data for any analysis, it is important that more realistic data collected from real life situations is used. Unfortunately, the use of real world data can have legal or moral issues, and thus many real-world datasets can not be used for academic research purposes [12]. As a consequence the main datasets used in the area of SER for academic research work are IEMOCAP [2] and MELD [1]. These artificial datasets are described below.

2.9.1 IEMOCAP

The Interactive Emotional Dyadic Motion Capture Database (IEMOCAP) [2] is a database collected by the Speech Analysis and Interpretation Laboratory at the University of South California. It was recorded from ten actors, both male and female, in dyadic sessions with the actors augmented with markers on their face, head and hands. The database contains both scripted and improvised sessions. For each improvised and scripted recording, the dataset provides detailed audiovisual and text information, which consists of audio and video of both interlocutors, motion capture data of the face, head and hand of one of the interlocutors in each recording, text transcriptions of the conversation, and their word-level, phone-level and syllable-level alignment. The dataset contains approximately 12 hours of data. The audio is split into segments of between 3-15 seconds and each segment is labeled by 3–4 human evaluators. Audio utterance lengths vary from 0.6 seconds up to 34.1 seconds, with almost 80% of the samples being 6 seconds or less. The database was initially designed to target the emotions of anger, sadness, happiness, frustrated and neutral state. The

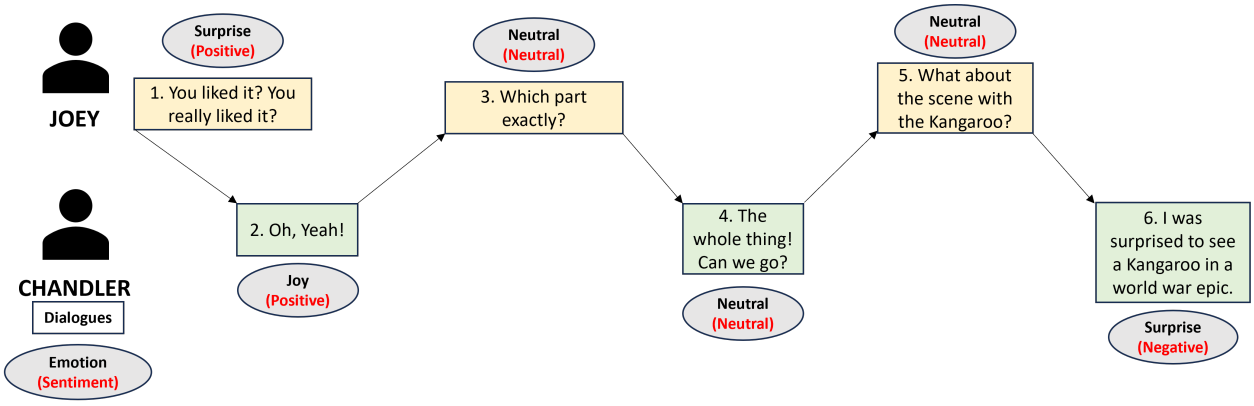


Figure 2.14: Example dialogue of MELD dataset. Redrawn from [1].

database was expanded later to code for 10 emotion categories; the emotions of anger, sadness, happiness, disgust, fear, surprise, frustration, excited, neutral, and other. It is a well-known database and has been extensively studied. In this work we utilize the audio utterance level data in the IEMOCAP dataset.

2.9.2 MELD

MELD [1] is a multimodal emotion/sentiment classification dataset which has been created by enhancing and extending the EmotionLines dataset [3]. In contrast to the IEMOCAP dataset, MELD is a multiparty dialog dataset and has almost twice the number of utterances as found in IEMOCAP dataset. MELD contains textual, acoustic and visual information for more than 1400 dialogues and 13000 utterances from the TV series "Friends" [147]. Each utterance in every dialog is annotated with one of the seven emotion classes: anger, disgust, sadness, joy, surprise, fear or neutral. For each utterance text, audio, and visual information associated with the utterance is available. In this work we utilize only the audio portion of the dataset. Figure 2.14 shows an example dialogue from the MELD dataset.

2.10 Emotional-informed dialog systems

The last few years has witnessed a significant amount of research in the area of emotion-informed dialogue systems. A number of methods have been proposed to fulfill the task of generating and rendering emotionally infused responses. This infusion aims to create systems that can provide maximum user engagement and more natural and effective communication in a human-computer interaction [148]. Once an emotion recognition system is designed it can be incorporated with a dialogue generation system to provide a complete end-to-end emotion aware dialogue system. There already exists work in this area. For example, [149] proposes a novel effective dialogue system "Mixture of Empathetic Listeners (MoEL)" that incorporates empathy in an end to end approach for generating responses. This model learns and responds to each emotion presented by the user by learning specific listeners for each sensed emotion. MIMicking Emotions for Empathetic Response Generation (MIME) [150] is an interactive system based on the assumption that emphatic responses should mimic the emotion of the speaker to a certain degree. This model tries to balance the response by generating positive responses for positive utterances and for negative utterances generate a response that incorporates a composite emotion that agrees with the user and to generating responses that are positive and more hopeful. [151] describes the use of an interactive model that incorporates the user's feedback when generating responses that are aligned with the dialogue's history and user's estimated emotion. This model shows improved performance as compared to earlier systems in generating emphatic dialogues.

Research in emotional conversation systems (e.g [152], [153]) focuses on building emotion-controllable conversation systems based on the estimated user-presented emotion. These systems apply various mechanisms to infuse a specific emotion vector into the response generation process and use this to enhance the computer's emotional expression. Historically, these systems are based on text input in isolation and ignore potential information from facial expressions, audio, speakers' personalities, and the emotional trajectory of an interaction or conversation. More recent work has moved from considering lines of text in isolation to systems that use multi-modal cues to emotional state as well as considering the time course of emotion in a

conversation. For example, emotion can be established based on visual appearance [35], audio propagation of speech [154], and body appearance and gestures [155]. Other work has begun to explore tracking the emotional state in dialogue history when generating emotional responses. For example, [156] proposes a neural network-based approach for positive affect-sensitive response generation. While [157] proposes a Hierarchical Recurrent Encoder-Decoder (HRED) model, to capture the emotional context of a dialogue. They concluded that their architecture resulted in a better model and that it produces responses that are perceived as being more natural and eliciting a more positive emotional impact on the user. Their model also generates shorter responses that contain more positive-sentiment words. This approach resembles a commonly used human strategy when promoting positive emotion in a conversation with limited context [158].

Large Language Models (LLM's) are gaining increasing popularity because of their outstanding performance in various applications including translation, text generation, transcription, cybersecurity and many more. There is ongoing research work that uses LLM models such as ChapGPT for creating consistent emotion annotations [159]. One recent paper reviews LLM's capacity in demonstrating empathy [160]. Based on their review they conclude that generally, existing LLM's lacks emotional capability. This may be an interesting direction for future research.

2.11 Summary

Emotion Recognition In Conversation (ERC) is a popular task in the Natural Language Processing (NLP) community with the availability of online conversation data from social media websites such as Instagram, Facebook, Reddit, YouTube, and many more. The fundamental task is to recognize emotions from utterances in a conversation. Many models have been proposed in this area that have tried to model the emotional complexities using speaker information, topic of conversation, contextual information and many more factors into account. But most systems deal with text data only. The problem with using textual data is that the text does not necessarily capture the emotion being expressed. The actual speech can contain more sophisticated

and complete information. Speech is complex data that contains information about speaker, gender, message context and for our task, most importantly emotions. To explore this, in this work, we concentrate on the speech signal as the input data for performing emotion recognition.

Various models exist for speech based emotion recognition. Early work used traditional techniques for speech emotion recognition including Support Vector Machine (SVMs), Gaussian Mixture Models (GMMs), and Hidden Markov Model (HMM). These early approaches have been generally replaced by deep learning models. There exists a number of deep learning techniques including CNNs, RNNs, MLPs, and autoencoders, thus can be adapted to SER. However, more recently graph-based neural network approaches have proven quite successful for speech-based emotion recognition as they provide a mechanism to relate the various constraints in ERC. A GNN-based approach is considered here.

Most of the models for speech emotion recognition in the literature use text transcriptions of audio utterances to train the model for later recognition. Typically, the models are non-order and non-speaker specific although models such as DialogueGCN do include these properties. This thesis adapts the text-based DialogueGCN model described in [33] to use acoustic information rather than text. As with DialogueGCN the model takes into account speaker dependency and contextual information related to the text utterances for emotion detection in conversations. In this thesis I seek to answer the question "Can the performance of ERC in conversational settings be improved when speaker information, contextual information along with speech utterance features are included in the model by incorporating a graph-based neural network methodology?" I begin to answer this question in the next chapter.

Chapter 3

Methodology

Speech emotion recognition (SER) is the task of automatically recognizing emotions expressed in spoken language [161]. Current approaches to SER focus on analyzing isolated speech segments (utterances) to identify a speaker's emotional state [29, 162]. A major challenge in SER task is that emotion-relevant information is scattered throughout into different parts of the conversation, therefore an efficient modeling capacity for long-term context is needed [22]. Speech emotion recognition in conversation (SERC) attempts to identify and detect human emotions by capturing and characterizing emotion attributes in speech utterances within conversational context rather than in isolated utterances [33].

In this work, to perform SERC we have adapted and enhanced the current state-of-the-art GNN-based ERC model [33], originally proposed for text-only input, to spoken dialogues. The model operates on text utterances as input and uses a bidirectional GRU to extract the utterance-level features. Further, it uses a graph-based approach to model speaker-dependent contextual information in a conversation. In our work, we refine the audio utterance encoding which requires processing, operating, and modeling audio data to suit

the speech-related features. Moreover, we use a graph-based approach to transform each individual audio utterance into a graph using GCN to extract the utterance-level features. The GCN-based modeling allows our framework to handle audio data and complex dialogue interactions more effectively.

The model of Ghosal et al. [33] suffers from graph complexity and due to this the model suffers from learning meaningful speaker dependencies. To overcome this limiting factor we address the issue through *graph sparsification* [39]. We only consider edges that occurred within a small temporal window, (i.e., the window goes symmetrically forward and backward in time based on the window size) to account for the importance of recency of emotions in dialogues. The sparsification method eliminates a large number of edges from the original complete graph, while maintaining the most recent ones (recency) [40]. In addition, we preserve a speaker's relation to their own utterances by maintaining all same-speaker edges in the graph, to account for the emotional persistence and maintaining long-term emotional profile of the user (self-dependency). As a result, with our approach, we reduce the graph complexity and also manage to accommodate recent and relevant edges and model *recency* and *same-speaker dependency*, which are essential factors in ERC [33, 35].

3.1 Formal problem definition

Consider a sequence of generative audio utterances $(u_1, u_2, u_3, ...u_N)$ from S speakers $\{s_1, s_2, ..., s_S\}$ where $(S \geq 2)$. The SERC task is to assign an emotion label $l(u_i) \in L = \{l_1, ...l_y\}$ to the constituent speech signal utterances $(u_1, u_2, ..., u_N)$, using context and speaker information from neighbouring utterances. Specifically we seek to identify the emotion label $l(u_i)$ for each target utterance. Here, L is a set of predefined emotion labels. Different sets of emotions labels may be used depending on the conducted experiment. We assume that the speaker of each utterance is known.

3.2 Basic approach

Our approach to the SERC problem aims at adopting the current state-of-the-art GNN-based ERC model [33], originally proposed for text-only input, to spoken dialogues, and in our work use it for audio input. The utterance-level feature extraction process from the input audio utterances follows a graph-based approach. As described in [31] the input audio utterances are first transformed into cyclic graphs, which then pass through GCN layers and a pooling layer to extract the utterance level features. The methodology described in [33], suffers from graph complexity which we have addressed in our approach. The key idea is that we address the graph complexity and its effect on learning meaningful speaker dependencies by *graph sparsification* [39]. This is done by only considering other speaker edges that occurred in a smaller temporal window, (e.g., connecting each node with edges with a small context window). The graph sparsification effectively eliminates a large number of edges from the original complete graph, by maintaining the most recent ones. This makes the approach more computationally efficient. In addition, we preserve a speaker's relation to its own utterances by maintaining all same-speaker edges in the graph. As a result, the graph-based approach for utterance level feature extraction, incorporating *recency* and *same-speaker dependency* (or *self-dependency*) helps in building an efficient approach for performing speech emotion recognition in conversation. The SERC-GCN model is illustrated in Figure 3.1. SERC-GCN is built up of two components, a conversation-independent independent (Stage One) and conversation dependent stage (Stage Two). These are discussed in detail below.

3.3 Stage One: Context-free

As described in Chapter 2, speech emotion recognition is usually comprised of three fundamental steps; preprocessing of the speech signal, speech feature extraction and classification. To extract the speech features from the raw audio signal is extremely important as then these features are fed to a machine learning model

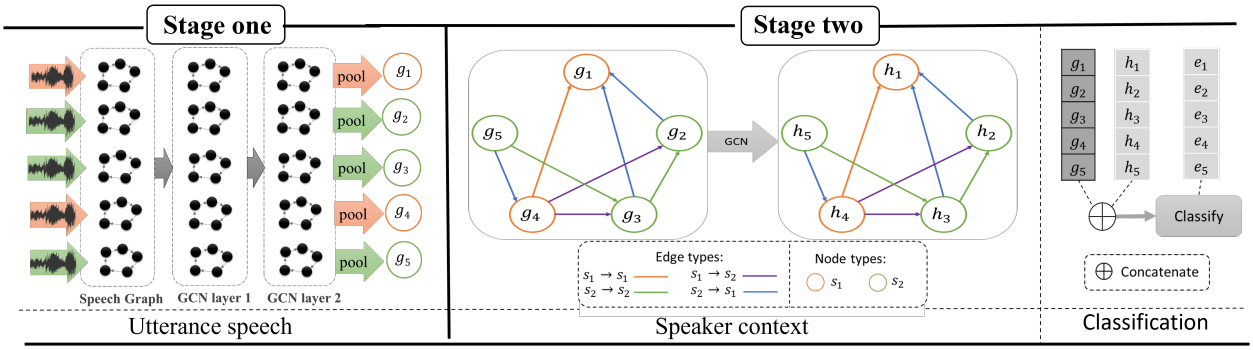


Figure 3.1: Overview of the SERC-GCN model's architecture for speech emotion recognition. The algorithm consists of two stages. In Stage One individual audio utterances are encoded as cyclic graphs and then passed to two GCN layers followed by a pooling layer to extract the utterance level features (g1,g2,..,g5). These features are then used to initialize each node in Stage Two for constructing a conversation graph. A feature transformation method is applied that transforms the utterance level features into speaker-dependent features (h1,h2,..,h5) as shown. Finally, each utterance is represented as the concatenation of g_i and h_i features passed through a softmax layer for emotion classification.

(deep learning model or graph-based model) to generate discrete emotion labels for each input utterance. For our work, we have followed a graph-based approach proposed by Shirian and Guha [31] to extract speech utterance features. In their proposed method speech emotion recognition is performed using a graph classification approach. As shown in Figure 3.2, each input speech samples/utterances are first transformed into graphs where each node in the graph corresponds to a short windowed segment of the signal. Developing these graph structures helps enable the graph-based GCN architecture to perform a more accurate and efficient graph convolution as compared to the approximate convolution performed in the standard GCN architecture. Next, a GCN architecture is developed with two GCN layers followed by a pooling layer to obtain graph-level embeddings (latent representation) and then finally passed to a fully connected network and a softmax layer to assign emotion labels to each speech utterance for training.

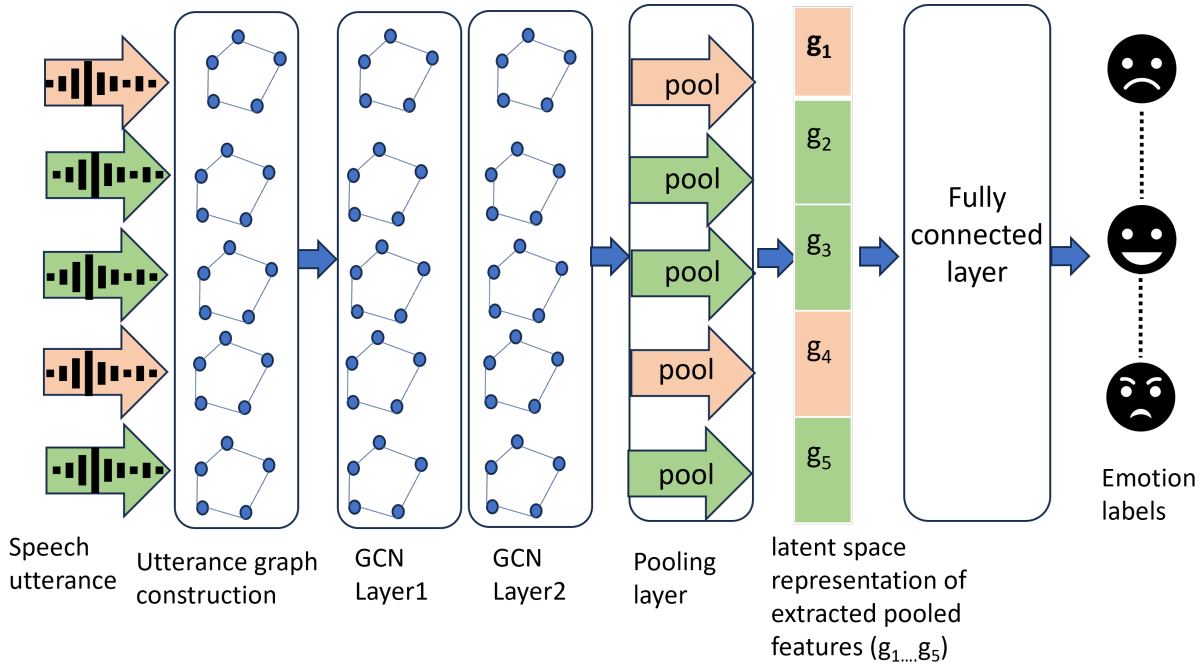
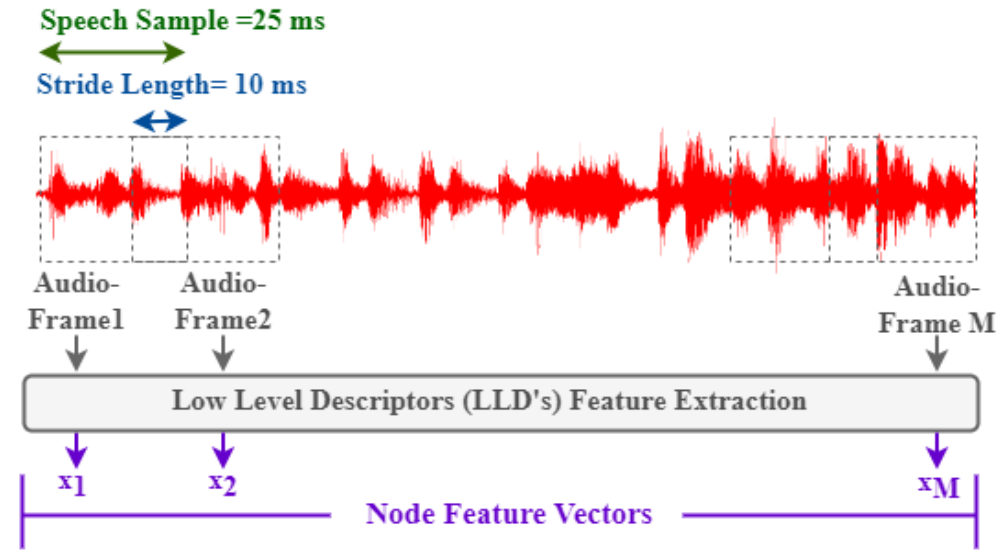


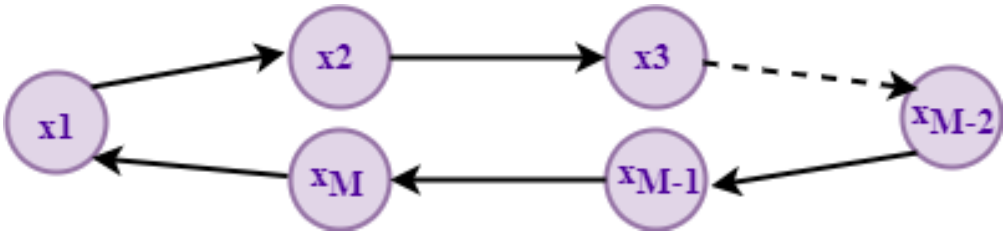
Figure 3.2: Graph-based architecture for speech emotion recognition. Adopted from [31]. $[g_1....g_5]$ is the latent space representation of the labelling prior to presenting the data for labelling. This architecture is adopted for Stage One to extract the utterance-level features from the pooling layer.

We have adopted this methodology for our work and modified it to extract the utterance level features from the pooling layer instead of the fully connected layer which is last layer in the graph-based architecture [31]. The extracted features from the pooling layer represent the speech utterance level latent space representation (g_i) in our methodology for Stage One. The detailed steps involved in the extraction process are described below. We choose this feature (g_i) as it reduces the complexity of the neural network while it provides an effective latent representation as will be shown shortly.

In Stage One the input audio utterance is modeled as a graph, where each vertex corresponds to a short windowed segment of the audio signal. Each vertex is connected to its two adjacent vertices transforming the audio signal into a cyclic graph. The details of this process are shown in Figure 3.3 and are described



(a) Audio frame to node transformation.



(b) Cyclic graph construction.

Figure 3.3: Transforming speech audio to an utterance graph.

below.

Utterance Graph Construction: Modeling a single speech utterance as a graph follows a frame-to-node transformation, where M frames (short, overlapping segments) of the speech signal form the M vertices in graph G_u. As shown in Figure 3.3a to extract the low-level descriptors (LLDs) from the raw audio utterance, OpenSMILE toolkit [126] is used. The feature set used is based on the collection of standard acoustic measures and includes Melfrequency cepstral coefficients (MFCCs), zero-crossing rate, voice probability, frame energy, and fundamental frequency (f0). Each speech utterance is constructed as a directed cyclic graph $G_u = (Z_u, E_u)$, where $Z_u = \{z_1, z_2....z_M\}$ is the set of M vertices of speech utterance and the set of edges is defined as

$$E_u = \bigcup_{i=1}^{M-1} \{(z_i, z_{i+1})\} \cup \{(z_M, z_1)\} \tag{3.1}$$

Mapping an utterance to its LLDs involves processing its audio signal with a sliding window and extracting measurements. Similar to [31], each vertex z_i is associated with a vertex feature vector x_i which are the LLDs of the vertex features dimension of $|x_i| = 35$. Each speech sample produces a graph of $M = 120$ vertices where each vertex corresponds to an overlapping speech segment of $25ms$ with a stride length of $10ms$, corresponding to a speech sample length of $1.81s$. In order to deal with variable-length speech utterances padding and truncation are used to make utterances of equal length. This makes all the graphs of equal length with 120 nodes. These values are selected to be consistent with the approach taken in [31]. The process is shown in Figure 3.3b.

Utterance feature extraction

Each speech utterance forms a directed cyclic graph. This graph passes through the GCN architecture with two convolutional layers to extract speech features of the utterances. The convolutional operator used in the GCN architecture is spectral convolution. Following the convolution operation [31] the information from the neighboring nodes are aggregated and then passed into a pooling layer. The pooling layer provides a graph-level representation (p_G) of the utterance from the node-level embeddings (N^k). In the model architecture, the pooling layer is employed on the node embeddings after the last GCN layer to obtain graph representation.

$$p_G = pooling function(N^k) \tag{3.2}$$

Three pooling functions are in common use in the literature, max pooling, sum pooling, and min pooling.

- Max Pooling: Pools a graph by computing the maximum of its node features.
- Sum Pooling: Pools a graph by computing the sum of its node features.

- Min Pooling: Pools a graph by computing the minimum of its node features.

In order to identify the most appropriate pooling function, we extracted the utterance-level features with each of these pooling functions. Details of this experimentation are presented later in the chapter. Out of these pooling functions, sum pooling resulted in higher weighted percent accuracy ($WA\%$) accuracy.

$$p_G = sumpool(N^k) \tag{3.3}$$

The extracted pooled features (g_i) are the output from Stage One. These context-free speech utterance features (g_i) are used to initialize the nodes of the conversation graph in the next stage.

3.4 Stage Two: Context-dependant

In this stage, the conversation graph is constructed using the information provided by Stage One. The context-free utterance level features (g_i) extracted from Stage One are used to initialize each vertex in the conversation graph.

3.4.1 Conversation graph construction

As shown in Figure 3.1 we construct each conversation from a sequence of N utterances $(u_1, u_2, u_3, ...u_N)$ as a directed graph $G_c = (V_c, E_c, R_c, W_c)$. Each utterance is represented as a vertex $v_i \in V_c$ and is initialized with the utterance speech feature vector g_i, $i \in [1, 2, ..., N]$, obtained from the individual utterances in Stage One.

For SERC-GCN model, two important modelling strategies are incorporated in the model:

Recency Modeling: Most conversation-based models assume that considering all of the conversation during analysis plays an important role in achieving a higher performance. However, the nature of an evoked emotion might be triggered by recent turns of the conversation [40]. Does the model rely on all utterances of the conversation or focus on the more recent ones? To address this question, we performed an ablation study to explore the temporal window dimension w. Specifically, we varied the length of the temporal window w that controls how far the window extends backwards and forwards in time. By varying the temporal window length w for both directions we can identify the window that provides the best result for the emotion recognition task. Detailed results of this study are provided in the next chapter.

Self-dependency Model: Deep learning models presented in the literature to date do not include speaker dependency. However, the nature of an evoked emotion might be dependent on the way a specific speaker converses and it becomes important to capture speaker-dependent contextual information in a conversation. In conversational dynamics two types of speaker dependency are important to consider: (i) inter-speaker dependency and (ii) self-speaker dependency. Interspeaker dependency encodes the emotional influence that one speaker has or produces on another speaker. Generally, this dependency is related to the fact that speakers build rapport during the course of a conversation and this is reflected in the emotion expressed by the speaker. But it may be the case that one speaker does not have any effect on another speaker. Self-speaker dependency, or emotional persistence, deals with the aspect of emotional influence that speakers have on themselves during conversations. Speakers in a conversation are likely to stick to their own emotional state due to their emotional inertia unless the other speaker invokes a change.

3.4.2 Relational edges construction

Each vertex v_i representing an utterance in the conversation has directed edges connecting past and future utterances. Within some temporal window an ablation study is conducted to find the window size for connecting past and future utterances and preliminary experimental results showed a window of three utter-

ances performs best (for IEMOCAP), as it preserves recency, a known factor in emotional dynamics. The rest of the edges are omitted edges that extend beyond the conversation which makes this approach more computationally efficient. Same user utterances are also connected through directed edges. The relation r of an edge r_{ij} is set based on the speaker dependency between speaker s_i (of utterance v_i) and speaker s_j (of utterance v_j). We use this relation to represent the past (backward) and future (forward) temporal dependency between the vertices. The edge weights are set using a similarity-based attention module. The attention function is computed in a way such that, for each vertex, the incoming set of edges has a sum total weight of 1. The weights are calculated as follows:

$$\alpha_{ij} = softmax(v_i \odot W_j) \tag{3.4}$$

v_i is the vector of edges connected to $vertexi$, and W_j is a vector of weights assigned to each edge. $\odot$ is the Hadamard product. The use of the softmax function in the classifcation ensures that V_i receives a total weight contribution of 1.

3.4.3 Feature transformation method

This transformation method transforms the extracted utterance level features which are speaker independent into speaker dependant using a graph-based approach similar to [33]. The utterance speech features g_i are transformed from speaker-independent into speaker-dependent feature vectors h_i. In the first step, a feature vector is computed for each vertex, by aggregating local neighborhood information. In our case, the neighbor utterances are specified by the past and future window size:

$$h_i = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{\alpha_{ij}}{c_{i,r}} W_r \, g_j + \alpha_{ii} W_0 \, g_i), \tag{3.5}$$

for $i = 1, 2, ..., N$, where α_{ii} and α_{ij} are the edge weights, and N_i^r is the neighbouring indices of vertex i under relation $r \in R_c$. $c_{i,r}$ is a problem-specific normalization constant automatically learned in a gradient-based learning setup. σ is an activation function such as ReLU, and W_r and W_0 are learnable parameters of the

transformation. In the second step, another transformation is applied to accumulate the normalized sum of the local neighborhood:

$$h_i = \sigma(\sum_{j \in N_i^r} W\, h_j + \alpha_{ii} W_0\, h_i) \tag{3.6}$$

for $i = 1, 2, ..., N$, where W_c and W_0 are transformation parameters, and σ is the activation function.

The above transformations accumulate the local neighborhood information for each utterance in the conversation graph. This method is adopted from [33].

3.5 Emotion classifier

After applying GCN feature transformation on the conversation graph, emotion classification is performed. To obtain the final speech utterance representation, the utterance level feature vectors (g_i) from Stage One and the context speaker-dependent feature vectors (h_i) from Stage Two are concatenated. This final utterance representation is then passed through a softmax layer to be classified into one of the emotion labels.

3.6 Training

SERC-GCN is a two-stage process. Below, we describe the training process for Stage One and Stage Two. Stage One is context-independent and is trained at the audio utterance level with emotion labels and utterance-level features are extracted from the pooling layer at the end of Stage One. We evaluate the various pooling options and extract the utterance-level features from each of these functions and try to find the pooling function that achieves the highest accuracy for Stage Two. For Stage Two, the training is done using conversation along with speaker dependency and recency to build relational edges. The training process is based on supervised learning. As all the datasets have different emotion labels chosen, training is performed separately for each dataset.

Table 3.1: IEMOCAP dataset statistic with number of utterances for each emotion class

Emotion label	**Number of Utterances**
"xxx"	2500
frustrated	1849
Angry	1103
Happy	595
Neutral	1708
Sad	1804
Excited	1041
Surprise	111
Fear	40
Other	30
Disgusted	20

3.6.1 Stage One training

In Stage One, input audio utterances are transformed into a cyclic graph. Each audio utterance graph is comprised of 120 nodes where each node corresponds to an audio utterance (sample length) length of 25ms with an overlapping length (stride length) of 10ms. All of the audio samples are of equal length. Each node is associated with a feature vector of dimension 35 for both the IEMOCAP and MELD datasets. For the IEMOCAP dataset, five emotion labels are provided for Set-1 including *angry*, *happy*, *neutral*, *sad* and *excited* and for Set-2 including *angry*, *happy*, *neutral*, *sad* and, *frustration*.

3.6.2 IEMOCAP dataset description

The Interactive Emotional Dyadic Motion Capture Database (IEMOCAP) [2] dataset contains data from 10 actors both male and female involved in a dyadic conversation. The database contains both improvised and scripted sessions. Generally, considering only the scripted session can result in questionable results as these scenarios do not represent everyday life. To increase the sample size we have used both sessions during model training. The full dataset contains 10,039 utterances, with complete emotion class statistics is described in Table 3.1. All these samples are split up equally in five sessions including each male and female participant in each session. Each utterance in the dataset is manually annotated by atleast three annotators with an emotion label. In IEMOCAP, the label "xxx" identifies samples for which there was no agreement among the annotators. In contrast "other" identifies that the annotators agree on some emotions but it is not a part of the predefined list of the dataset. So in our analysis, we did not consider "xxx" and "other" label. Also, we did not consider emotion labels: 'surprised", "fearful", "disgusted" as these emotion lables have too small number of utterances. To be consistent with the baseline models we choose to explore the dataset with the same emotion categories as listed as Set-1 and Set-2.

3.6.3 Balancing the IEMOCAP dataset

To be consistent with the reference model [31], we considered five emotions from Ekman's emotion categories: *angry*, *happy*, *neutral*, *sad* and *excited*. The total number of audio utterances distribution is given in T̃able 3.2. A second experiment is carried out with another set of emotion categories and to be consistent with the reference model [23], we considered *angry*, *frustration*, *happy*, *neutral*, and *sad* to simplify the process of comparing our model with utterance only SER models. To make the dataset more balanced, the approach of merging the "excited" and "happy" classes into a single class as "happy" is adopted as common in the literature (e.g., [163], [164]). The dataset distribution is described in Table 3.3.

Table 3.2: IEMOCAP dataset statistic for Stage One for training and evaluation **(Set-1)**

Emotion label	**Number of Utterances**
Angry	1103
Happy	595
Neutral	1708
Sad	1804
Excited	1041

Table 3.3: IEMOCAP dataset statistic for Stage One for training and evaluation **(Set-2)**

Emotion label	**Number of Utterances**
Angry	1103
Frustration	1849
Happy	1636
Neutral	1708
Sad	1804

Table 3.4: IEMOCAP dataset sampling for training the graph model.

Dialogues			**Utterances**		
train	val	test	train	val	test
108	12	31	5163	647	1623

For training the model the distribution of the dataset used for both the stages is shown in Table 3.4.

3.6.4 MELD dataset

MELD [1] is a multimodal emotion/sentiment classification dataset that has been created by extending the EmotionLines dataset [3]. The EmotionLines dataset contains the dialogue from the T.V series Friends [147]

Table 3.5: MELD dataset acoustic component with the number of utterances for each emotion class

Emotion label	Number of Utterances
Anger	1606
Happy	2308
Neutral	6436
Sad	1003
surprise	1616
Fear	358
Disgust	361

where each dialogue has utterances from multiple speakers. When creating this multimodal dataset the authors enforced constraints including (i) the timestamps associated with the utterances in the dialogues must be in increasing order and (ii) all the utterances in the dialogue must belong to the same scene and episode. Incorporating these constraints reduced the number and nature of the ambiguities present in the dataset. Once the timestamps of each utterance are obtained then the audio-visual clip from the source episode is extracted and audio information is extracted from these clips. In this way, the final dataset contains audio, visual, and textual information related to each utterance. To create the MELD dataset three annotators labeled each utterance where majority voting decides the final label. The MELD dataset contains textual, acoustic, and visual information for more than 1400 dialogues and 13000 utterances from the Friends TV series. Each utterance in every dialog is annotated with one of the seven emotion classes: *anger*, *disgust*, *sadness*, *joy*, *surprise*, *fear* or *neutral*.

3.6.5 Balancing the MELD dataset

As shown in Table 3.5, the MELD dataset is highly unbalanced where the highest number of utterances belong to the neutral class, with total number of utterances $n = 6436$ while for other emotion categories

Table 3.6: MELD dataset sampling for training the graph model

Dialogues			**Utterances**		
train	val	test	train	val	test
1039	114	280	9989	1109	2610

including fear and disgust have much fewer utterances $n = 358$ and $n = 361$ in the distribution. In order to address this issue, we have used class weights in order to balance the dataset.

For training the model on MELD dataset we use the same train/test/val split as used by the baselines methods [34]. The dataset distribution is shown in Table 3.6.

3.6.6 Model implementation and hyperparameters

To measure loss during training Stage One Cross entropy is used as the error function.

$$loss = -\sum_{n} y_n log(y_n) \tag{3.7}$$

During training, the sample graphs are fed to a two-layered GCN architecture followed by a pooling layer and a fully connected layer with a softmax layer at the end to produce the classification labels. After the network is trained, we extract the features from the pooling layer which is one step ahead of the fully connected layer. We ran the model for 1000 epochs with the Adam optimizer. The output of Stage One are the pooled feature with a dimension of 128. Early stopping is performed after each 50 epochs with a rate of 0.5. Utterance features are extracted from the best checkpoint. Pytorch geometric an open source Python library is used for the experiments [165]. The details of the hyperparameters used for training are listed in Table 3.7.

To compute graph-level representation different poling strategies are attempted; mean, max, and sum pooling. For each of the pooling layer, the above process is repeated. In the end, utterance level features

Table 3.7: Hyperparameters used for Stage-One training

Hyper-parameters	**Value**
Batch size for training	32
Learning rate	0.01
Epochs	1000
Epochs per iteration	50
Fold	5
Drop out	0.5
Weight Decay	1e-4
Number of GCN layers	2
Number of hidden units	64

are extracted from each different type of pooling layer to test stage two performance based on these pooled extracted features.

3.7 Comparing model performance using different pooling strategy

We have compared model performance using three different pooling strategies. As reported in Table 3.8, we found that our model SERC-GCN performs best using sum pooling layer in Stage One and reported highest performance with 66.85% accuracy for Set-1, 51.5% accuracy for Set-2 of IEMOCAP dataset, and for the MELD dataset it achieved 51.7% accuracy. This shows an improvement of 0.52% over maxpool and 0.36% over meanpool for IEMOCAP Set-1 dataset. For Set-2, sum pooling shows an improvement of 1.37% over max pooling and 0.38% over mean pooling. For the MELD dataset sum pooling shows an improvement of 1.57% over max pooling and 0.77% over mean pooling. Sum pooling layer-based utterance level proves to be the best for extracting features in Stage-One for all three datasets.

Table 3.8: Different pooling strategies (WA%) evaluation on IEMOCAP (Set-1 and Set-2) and MELD datasets

Pooling type	**IEMOCAP Set-1**	**IEMOCAP Set-2**	**MELD**
Max Pool	66.5	50.8	50.9
Mean Pool	66.61	51.3	51.3
Sum Pool	**66.85**	**51.5**	**51.7**

3.8 Stage Two training

For Stage Two conversations are considered. During conversation graph construction, the utterances are initialized with the Stage One pooled features. We refer to these features as context-free features. As Stage One did not consider speaker identity or dependency, so the extracted features are speaker and conversational structure agnostic. To incorporate speaker dependency, a new feature vector (known as a speaker-dependant feature) is computed for each vertex by aggregating over local neighborhood information. The feature transformation method as described in DialogueGCN [33] is used to perform the transformation which accumulate the local features of the speaker's neighborhood for each utterance in the graph.

3.8.1 Model implementation and hyperparameters

For each dataset, we train the model for 60 epochs using the Adam optimizer. The hyperparameters used for training are given in Table 3.9. After the feature transformation is done and passed through the GCN layers, the final utterance features are represented as the concatenation of context-free and speaker-dependent features that are fed into a fully connected layer with a Softmax layer for classification. As a measure of loss during training categorical cross entropy along with L2-regularization is used [33]. Details on training and

Table 3.9: Hyperparameters used for Stage Two training

Hyper-parameters	**Value**
Batch size for training	32
Learning rate	0.001
Epochs	60
Drop out	0.4
Number of GCN layers	2

evaluation for each of the datasets, along with an analysis of the efficacy of different temporal windows are provided in the following next chapter.

3.9 SERC-GCN labelling end-to-end example

Once both the Stage One and Stage Two components are trained, the entire algorithm is run on a given conversation to perform SERC. To illustrate this, here we provide an end-to-end description of the process of identifying the emotion labelling of a conversation. We begin by highlighting the process of labelling an individual utterance (Stage One).

3.9.1 Stage One

Consider the first audio utterance of speaker A "I don't think I can do this anymore" as shown in Figure 3.4. In this stage, this audio utterance is first transformed into a cyclic graph as illustrated in 3.5. The graph has 120 nodes with each node corresponding to an overlapping speech segment of length 25ms with a stride length of $10ms$ as shown in Figure 3.5. Each node is connected to its adjacent node to form the directed cyclic graph. Each node is associated with a feature vector X_i that is obtained by the local LLD's extracted by OpenSmile toolkit. These features are associated with the node feature vector X_i with a dimension of 35.

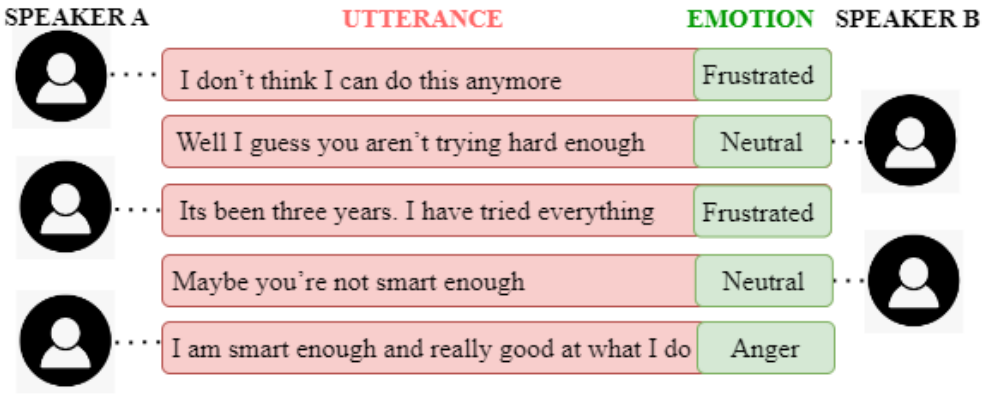


Figure 3.4: A turn-taking dialogue between two speakers A and B

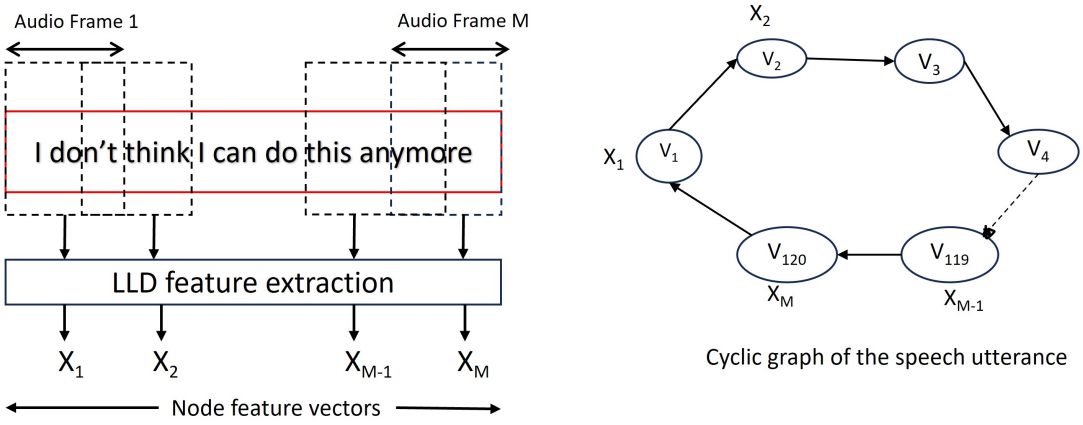


Figure 3.5: Stage One context-free speech feature extraction

After the graph is constructed, it passes through a GCN architecture with two GCN layers to obtain the node-level embeddings. Finally, it passes through a sum pooling layer to obtain the graph-level representation (latent representation). The extracted features from the sum pooling layer represent the speech utterance level latent space representation (g_i) as the output from Stage One. This process is repeated for each utterance in the conversation and the extracted features are later used to initialize the conversation graph in Stage Two.

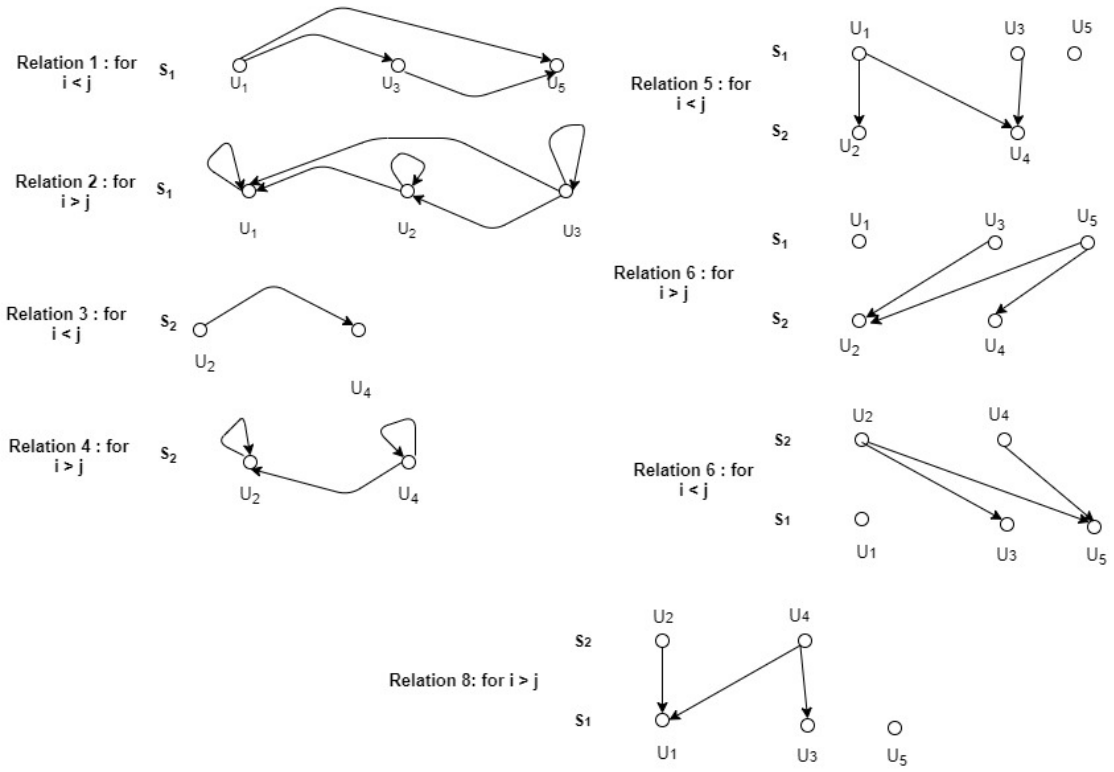


Figure 3.6: Edges Construction using from Table 3.10

3.9.2 Stage Two

The relational edges are constructed to capture context and speaker dependency among the different speakers within a conversation. For the purpose of this demonstration, assume that contextual information relies on all self-dependencies in the conversation, and a maximum temporal window of three utterances both forward and backward in time is adopted for this IEMOCAP dataset example conversation. A directed conversation graph is constructed where each node represents an utterance in a conversation and the edges are constructed based on the context to be modeled. Table 3.10 and Figure 3.6 summarize the edges presented in the graph. They are constructed both with same-speaker and other-speaker edges and are limited to the temporal window in use.

Once the conversation graph is constructed, it passes through two GCN layers. During the process, the

Relation Type	$s_s(u_i)$,$s_s(u_j)$	$i < j$	(i, j)
Same speaker/Temporal Dependency	s_1,s_1	yes	(1,3),(1,5),(3,5)
Same speaker/Temporal Dependency	s_1,s_1	no	(1,1),(3,1),(3,3), (5,1),(5,3),(5,5)
Same speaker/Temporal Dependency	s_2,s_2	yes	(2,4)
Same speaker/Temporal Dependency	s_2, s_2	no	(2,2),(4,2), (4,4)
Different speaker/ Temporal Dependency	s_1,s_2	yes	(1,2),(1,4),(3,4)
Different speaker/ Temporal Dependency	s_1, s_2	no	(3,2), (5,2), (5,4)
Different speaker/ Temporal Dependency	s_2,s_1	yes	(2,3),(2,5),(4,5)
Different speaker/ Temporal Dependency	s_2,s_1	no	(2,1),(4,1),(4,3)

Table 3.10: Edge relationship for two distinct speakers are redrawn from [33]. Here, $s_s(u_i)$, $s_s(u_j)$ denotes the speaker of utterances u_i and u_j. As there are two distinct speakers in a conversation, there can be maximum = $2M^2$ or eight distinct relation types r_{ij} in graph G.

feature transformation method is applied. This transformation uses the local neighborhood information to effectively accumulate the speaker-dependant features. Following this transformation step the context-free utterance features are transformed into speaker-dependent features. The final utterance feature representation is obtained by the concatenation of the extracted speech utterance feature vectors (g_i) from Stage One and the transformed speaker-dependent feature vectors (h_i) from Stage Two for emotion classification. A softmax layer is then used to assign final labels for each of the utterances in the conversation. For the example shown here the GNN assigns the labels *angry*, *happy*, *neutral*, *sad* and *excited*.

3.10 Summary

Chapter 3 describes the training process for Stage One and Stage Two of SERC-GCN which is customized for the specific dataset being used. Addressing the problem of SERC involves capturing the raw sentiment data from speaker utterances and then combining this information over the conversation to model the emotional content within the conversation. Building on previous work in SER, we formulate the problem in two stages. In Stage One, individual utterances are processed independently of the nature of the conversation to capture the raw emotional information in a given utterance and are transformed into cyclic graphs that describe a given utterance in terms of small overlapping temporal windows. This graph-based approach follows the model described in [31]. The speech sample graphs are then passed through two GCN layers followed by a pooling layer. The training processes described for Stage One demonstrate the extraction process of utterance-level features g_i from the different categories of pooling layer. Each time a pooling layer is selected (max, mean and sum) and features are extracted the same way as described. Then these features are used to initialize the nodes in the conversation graph of Stage Two.

In the second stage of SERC-GCN, raw utterance representations are used to construct a graph that represents the entire conversation. Nodes in this graph correspond to utterances which are initialized with the utterance level features from Stage One with edges encoding same-speaker and recency properties of

the conversation. This approach follows the model described in [33] for text-based ERC. In order to reduce the graph complexity of [33], we introduce the concept of temporal windowing to only encode temporal relationships between utterances within a smaller temporal window than the entire conversation. This more correctly encodes recency within the graph network which provides a more efficient SERC model.

.

Chapter 4

Experimental Evaluation

In this chapter, we report and discuss the results of the evaluation of our model. To show the effectiveness of our SERC-GCN model experiments are performed on on two benchmark datasets - IEMOCAP and MELD. IEMOCAP is considered in two subsets, based on the set of emotional labels used. The main reason of using these two datasets is to verify the methodology for a dyadic conversation using IEMOCAP and for a multi-party conversation using the MELD dataset. These datasets are multimodal datasets containing labelled textual, visual and acoustic information for every utterance of each conversation. We utilize the acoustic information from these datasets.

4.1 Evaluation criteria

To characterize/evaluate the predicted classification of an input utterance with the correct classification (the annotated label or ground truth of the utterance) for the task of emotion classification, the four metrics: true positives, false positives, true negatives, and false negatives are used. For example, consider an emotion

label (such as happy) as a positive class. If the classifier predicts the utterance belongs to happy class (i.e. the predicted label matches the annotated label), then the outcome is true positive (t_p). If the classifier incorrectly classifies the utterance belonging to the positive class, the outcome is false positive (f_p). If the classifier predicts the utterance belongs to the negative class but actually the utterance does belongs to the positive class (the annotated label belong to the positive class), then the outcome is False negative (f_n). And if the classifier predicts the utterance does not belong to the positive class and the annotated label of the utterance does not belong to positive class, then the outcome is true negative (t_n).

Based on the values of the confusion matrix, precision, recall as well as accuracy can be measured. These matrices are used to measure the performance of the model in multi-class classification problems.

Precision (π) measures how many utterances predicted by the classifier belong to the positive class (what is the percentage of the positive predicted utterances) and is defined as:

$$\pi = \frac{t_p}{t_p + f_p} \tag{4.1}$$

Recall (ρ) is also called as the true positive rate of the classification result. This means out of all the positive results predicted by the classifier, how many utterances actually belong to the positive class. It is defined as :

$$\rho = \frac{t_p}{t_p + f_n} \tag{4.2}$$

F-measure or **F1 score**, is measured using recall and precision and is defined as the harmonic mean of precision π and recall ρ. F1 score is calculated as:

$$F1score = \frac{2\pi\rho}{\pi + \rho} \tag{4.3}$$

The value of F1 score is maximum if $\pi = \rho$.

Table 4.1: Comparison of SER models and SERC-GCN on the IEMOCAP dataset, using weighted/unweighted accuracy (WA/UA). Best results are in bold. Second-best results are underlined.

Model	**WA** (%)	**UA**(%)
A. SER models		
Attn-BLSTM [36]	59.33	49.96
BLR [133]	62.54	57.85
RNN [134]	63.50	58.80
CRNN [135]	63.98	60.35
LSTM [136]	58.72	-
B. Graph SER models		
GCN [137]	56.14	52.36
PATCHY-SAN [139]	60.34	56.27
PATCHY-Diff [140]	63.23	58.71
Compact-SER [31]	65.29	62.27
Graph-SAGE [22]	<u>65.43</u>	<u>66.40</u>
SERC-GCN (ours)	**66.85**	**67.55**

The performance of a classifier can also be measured using accuracy, defined as the percentage of the correctly classified utterances w.r.t to the total number of utterances. Is is defined as:

Accuracy = Number of correctly classified utterances/ total number of utterances

- **Weighted Accuracy (WA%)**: It is computed by taking the average, over all the classes, of the fraction of correct predictions in this class.
- **Unweighted Accuracy (UA%)**: It is the fraction of instances predicted correctly (total correct predictions, divided by total instances).

Table 4.2: IEMOCAP Set-1 emotion labels using WA % (weighted average accuracy)

Emotion label	**WA** %
Happy	58.85
Sad	80.54
Neutral	66.78
Angry	68.87
Excited	64.18

4.2 Evaluation on the IEMOCAP dataset

For a comparative analysis of our SERC-GCN model performance, the following baselines for utterance-only models and conversational SER models are listed below. The utterances are annotated with one of the six emotions *happy*, *sad*, *neutral*, *angry* and *excited* and *frustrated* that are listed in the form of two sets (Set-1 and Set-2) for IEMOCAP dataset. As the dataset contains 5 sessions, we performed a 5-fold cross-validation (leave-one-out-session) and reported the weighted accuracy, unweighted accuracy and F1 score for the experiments. SERC-GCN results are reported for a window size of ($w = 3$).

4.2.1 Comparison with Utterance-only SER Models

IEMOCAP Set-1

Table 4.1 shows the results of utterance-only SER models trained on the Set-1 emotion categories: *happy*, *sad*, *neutral*, *angry* and *excited*. We report the weighted accuracy (WA%) and unweighted accuracy (UA%) of the models. We also report the results of our SERC-GCN model against nine other deep learning and graph-based SER baseline models. Our model outperforms all of these models. Specifically, S̃ERC-GCN

outperforms the second-best model, Graph-SAGE [22], by 2.17% and 1.73% for WA% and UA%, respectively. SERC-GCN also outperforms Compact-SER [31], which is the underlying method used in the context-free stage of SERC-GCN. Out of all the baseline models listed in Table 4.1 including the deep learning models Attn-BLSTM [36], BLR [133], RNN [134], CRNN [135], and LSTM [136] and some graph SER models GCN [137], PATCHY-SAN [139], PATCHY Diff [140], and Graph-SAGE [22] the algorithms's performance is provided in the author's paper. For the baseline model Compact-SER [31], the algorithm was implemented using the GitHub repo `(https://github.com/AmirSh15/Compact_SER)` provided by the authors. This model was also used for Stage One of this work. Training the model used the default hyperparameters as described in [31].

Table 4.2, summarizes the predicted emotion labels for the IEMOCAP Set-1 emotion classes. Performance is best on the Sad emotion class and worst on the Happy emotion class. Figure 4.1 shows the training accuracy and testing accuracy of our SERC-GCN model with the number of epochs. To achieve this testing accuracy the utterance level features are extracted in Stage One from the sum pooling layer. Then these extracted features are used to initialize the nodes in the conversation graph and edges are constructed with a window size of $w = 3$.

IEMOCAP Set-2

The utterance section in Table 4.3 shows the results of another set of utterance-only SER models trained on the IEMOCAP Set-2 of emotion categories: *angry*, *frustration*, *happy*, *neutral* and *sad*. These models report the Macro-F1 score unlike the models in Table 4.1. This is needed to do a comparative performance analysis of our proposed model and the referenced models. Table 4.3 also reports the results of our utterance-only SER model and our two-stage conversation SERC-GCN model. Both of our models outperform all other utterance-only SER models. For ResNet [38], BiLSTM [120], and Transformer [68] models the results are taken from the author's paper.

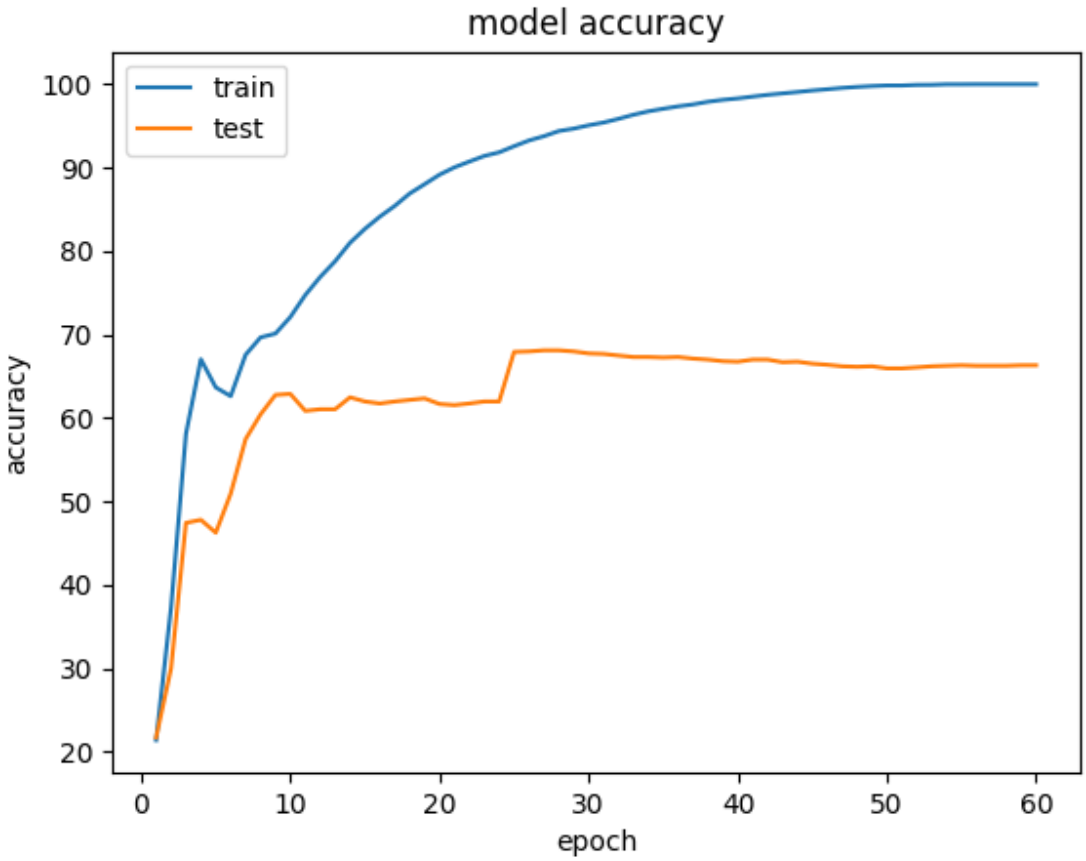


Figure 4.1: Training accuracy graph for IEMOCAP (Set-1)

4.2.2 Comparison with conversational SER models

The conversation section in Table 4.3 shows results for SERC models trained on the emotion categories Set-2: *angry*, *frustration*, *happy*, *neutral* and *sad*. Table 4.3 shows the effectiveness of our approach for the SERC task, where SERC-GCN outperforms all other SERC models. There is an improvement of 3.4% to the second-best result of a ResNet + transformer [23] model with 49.8% reported accuracy. SERC-GCN is more capable of modeling temporal and speaker information than existing SERC approaches. For Transformer [68] and ResNet + Transformer [23] models the results are taken from the author's paper.

Table 4.3: A comparison of SER models and SERC models on the IEMOCAP dataset, using Micro-F1.

Utterances	**Model**	**Micro-F1**
	ResNet [38]	34.3
	BiLSTM [120]	27.1
	Transformer [68]	39.1
Conversations	**Model**	**Micro-F1**
	Transformer [68]	45.3
	ResNet + Transformer [23]	49.8
	SERC-GCN (ours)	**51.5**

Table 4.4: IEMOCAP Set-2 emotion labels using WA % (weighted average accuracy)

Emotion label	**WA** %
Happy	47.28
Sad	60.21
Neutral	52.73
Angry	53.41
Frustrated	49.24

Table 4.4 summarizes for the IEMOCAP Set-2 emotion classes. Similar to Set-1, the performance on Sad is best while the performance on Happy emotion is the weakest.

Figure 4.2 shows the accuracy of training and testing the model against number of epochs. Once the classification model is trained on the Set-2 emotion categories the model is used to recognize labels for a test set and the resulting accuracy is shown in the graph. From the reported results using the IEMOCAP dataset and comparing our model performance with the baseline model for speech emotion recognition in conversations we can confirm that including (i) conversational context, (ii) temporal information, and (iii) speaker dependency (using recency to build the relationship edges between speakers) in a conversation

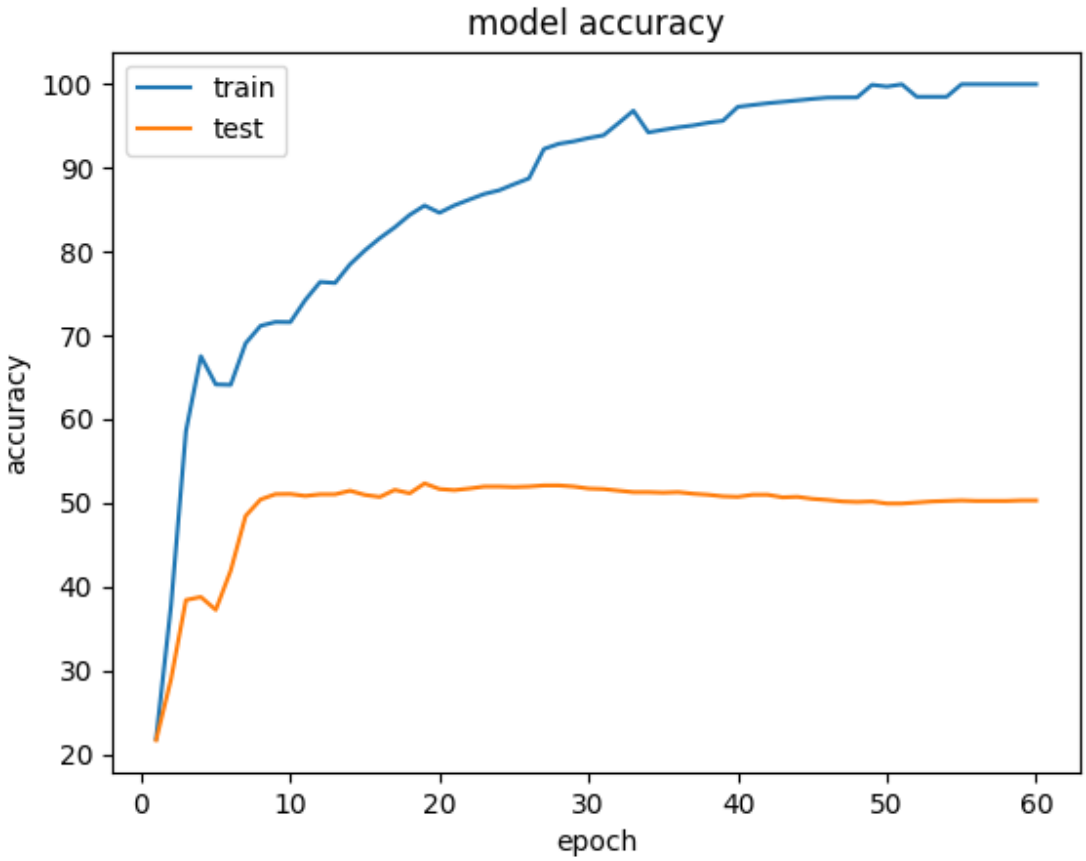


Figure 4.2: Training accuracy graph for IEMOCAP (Set-2)

graph, proves to be a much more efficient approach for speech emotion recognition in conversations. From the analysis of IEMOCAP dataset (for both sets), it is evident to say that incorporating context, speaker and temporal dependency information located in speech data can substantially improve the performance of speech emotion recognition.

Table 4.5: A comparison of SER models and SERC models on the MELD dataset, using WA % (weighted average accuracy)

Model	**WA** %
BC-LSTM [145]	36.4
CMN [35]	38.3
ICON [146]	37.7
DialogueRNN [37]	34.0
ConGCN [120]	42.2
Graph-attention [34]	<u>48.8</u>
SERC-GCN (ours)	**51.7**

4.3 Evaluation on the MELD dataset

4.3.1 Performance comparison with the baseline conversational models

For the multiparty MELD dataset, we compare our model SERC-GCN with the state-of-the-art baseline approaches for emotion recognition task in multi-speaker conversations. All the models listed use the MELD dataset for evaluation. For SERC-GCN a window size of $w = 4$ is used. Results from the baseline models listed in Table 4.5 including deep learning models BC-LSTM [145], CMN [35], ICON [146], and some graph models DialogueRNN [37], ConGCN [120], and Graph-attention [34] were taken from the results provided in the author's paper. Our model SERC-GCN, outperforms all of the other models baseline listed in Table 4.5 with a weighted average accuracy of 51.7%. This is almost 5.9% higher than graph-attention model [34] which is the second best model. The main reason for this improvement in accuracy as compared to the previous work [34] which uses a self-attention mechanism for utterance level feature extraction, is that our work uses a graph-based approach for utterance level feature extraction process. Feature extraction is extremely important in SER process as the extracted features that characterize the utterance directly

Table 4.6: MELD emotion labels using WA % (weighted average accuracy)

Emotion label	**WA** %
Neutral	80.1
Anger	50.3
Disgust	58.2
Sad	32.1
Joy	65.5
Surprise	23.6
Fear	50.8

influence the classifier performance and the overall efficiency of the model.

Table 4.6, presents the predicted emotion labels for the MELD emotion classes. Performance on Neutral is the strongest and Sad is the weakest. As in the IEMOCAP dataset, performance on Sad is the strongest while performance on Happy is the weakest. Figure 4.3 shows the training accuracy and testing accuracy of our SERC-GCN model with the number of epochs. To achieve this testing accuracy the utterance level features are extracted in Stage One from the sum pooling layer. Then these extracted features are used to initialize the nodes in the conversation graph and edges are constructed with window size $w = 4$ utterances in the conversation graph for emotion recognition.

4.4 Ablation study

The results above are based on the optimal recency window. Here we explain the sensitivity of the algorithm to the size of the recency window. With this study we have tried to answer the research question "Does the model rely on all utterances of the conversation or focus on the most recent ones?"

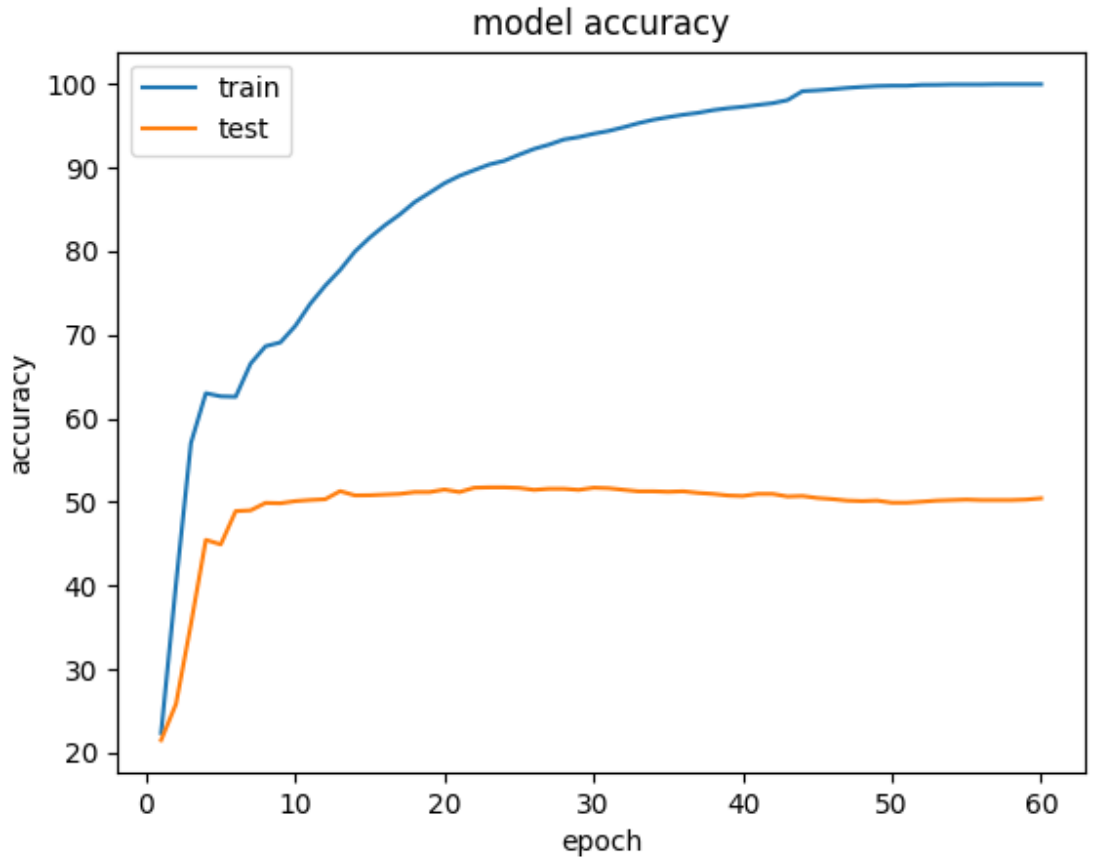


Figure 4.3: Training accuracy graph for MELD dataset

Table 4.7: Model performance analysis of the window w size for IEMOCAP AND MELD dataset

Dataset	$w = 1$	$w = 2$	$w = 3$	$w = 4$	$w = 5$	$w = 6$
IEMOCAP (Set-1)	65.9	66.1	**66.8**	65.9	66.3	65.5
IEMOCAP (Set-2)	50.0	50.2	**51.5**	50.3	50.1	51.0
MELD	50.1	50.3	51.2	**51.7**	51.1	50.9

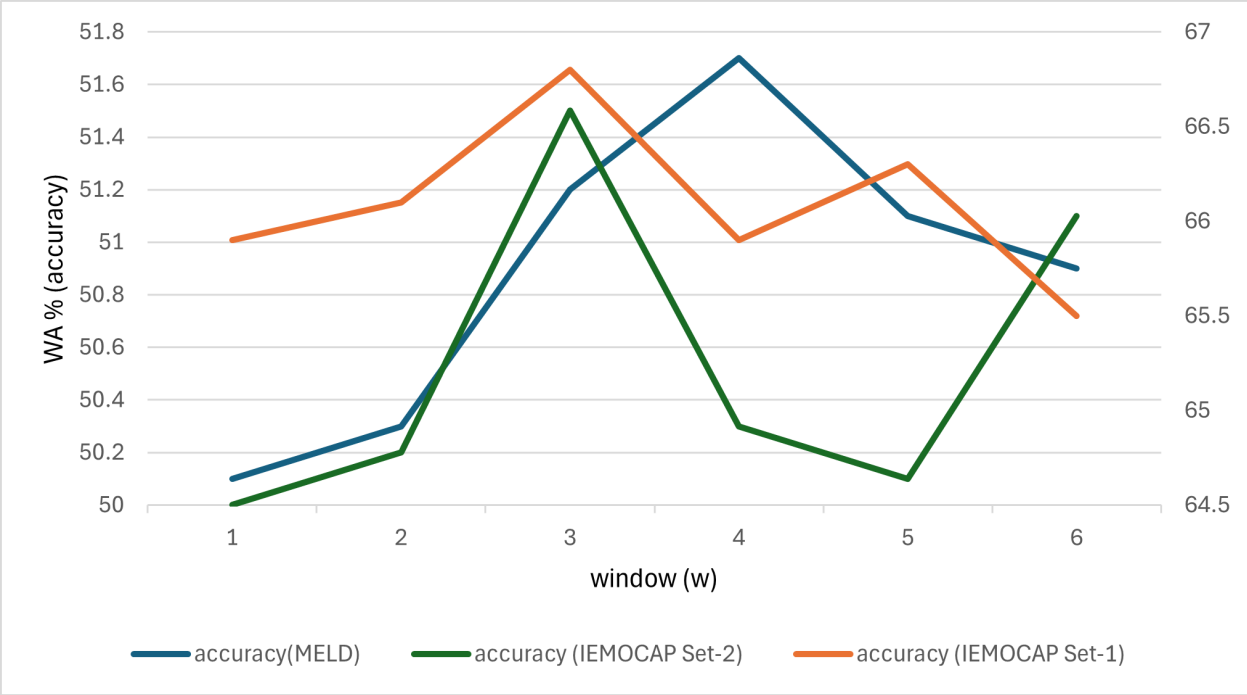


Figure 4.4: Weighted average accuracy (WA%) vs window parameter w

4.4.1 Size of the recency window (w)

In the Table 4.7 and Figure 4.4 show the accuracy (WA%) results as a function of the temporal window size (recency). The window parameter w varies both backward and forward in time. For example, with $w = 3$, each node (n_i) is connected with the n_{i-1}, n_{i-2}, and n_{i-3} edges backward and n_{i+1}, n_{i+2}, and n_{i+3} edges forward in-turn.

To identify the optimal recency window temporal window is varied. We carried out the experiments by varying temporal window from 1 to 6. We found that considering the most recent edges with window size $w = 3$ resulted in the highest performance IEMOCAP (Set-1) and IEMOCAP (Set-2) as compared to the other window values tested. The performance of the MELD dataset was also evaluated, and here a window size $w = 4$ was associated with best performance compared to the other window sizes evaluated.

Table 4.8: Performance comparison of conversation graph based on different model configurations for IEMOCAP dataset Set-1

Graph type	**WA%**
Complete graph	65.45
Conversation graph with recency without self-dependency	66.3
Conversation graph with recency and self-dependency	**66.85**

Table 4.9: Performance comparison of conversation graph based on different model configurations for IEMOCAP dataset Set-2

Graph type	**WA%**
Complete graph	50.2
Conversation graph with recency without self-dependency	51.1
Conversation graph with recency and self-dependency	**51.5**

4.4.2 Study based on edge relations by comparing recency and self-dependency

Here we verify the effectiveness of recency and the self-dependency in the conversation graph. Table 4.8 and Table 4.9 shows the performance comparison between a complete conversation graph where every utterance is connected to every other utterance, a conversation graph with recency ($w = 3$), and a conversation graph with recency and self-dependency for the IEMOCAP (Set-1) and IEMOCAP (Set-2) dataset. We found that highest accuracy, 66.85%, is achieved for Set-1 by incorporating recency and self-dependency into the conversation graph construction which is improved by 2.1% of complete graph and by 0.82% of a conversation graph with recency and without self-dependency. For Set-2, highest accuracy of 51.5% is achieved by incorporating recency and self-dependency into the conversation graph construction which is improved by 2.5% of complete graph and by 0.78% of a conversation graph with recency and without self-

Table 4.10: Performance comparison of conversation graph based on different model configuration for MELD dataset

Graph type	**WA%**
Complete graph	49.7
Conversation graph with recency without self-dependency	50.1
Conversation graph with recency and self-dependency	**51.7**

dependency. The complete graph means every utterance is connected to every other utterance and it shows the lowest performance because this graph has everything and due to this noise it lowers the performance. Similar results were achieved with the MELD dataset as shown in Table 4.10. Here a recency window of $w = 4$ was used. The highest accuracy is 51.7% which is achieved by constructing a conversation graph with recency and self-dependency. This configuration shows an improvement of 4.02% over a complete graph and a 3.19% improvement over a conversation graph with recency but without self-dependency. These results are strong evidence that incorporating recency and self-speaker dependency improves the performance of model for the emotion recognition task, even over the use of a complete graph.

4.5 Summary

In Section 4.2 we presented the result of our model SERC-GCN performance compared with other utterance-only and conversational baselines models. All these models use audio modality for the task of speech emotion recognition. For experimentation and comparison with the referenced models we have used two IEMOCAP sets using different emotion categories. For both Set-1 (happy, sad, neutral, angry and excited) and Set-2 (angry, frustration, happy, neutral and sad) our model outperforms the baseline model. In section 4.3 we presented the result of our model SERC-GCN performance compared with other popular conversational baseline models on the MELD dataset. Our model outperformed the baseline models. This could be attributed

to the fact that the baselines models did not consider temporal dependency and speaker dependency into account. Incorporating these factors into our models proved to be efficient for speech emotion recognition task. Another contributing factor for the performance improvement is that we only consider edges based on the temporal window size w. In Section 4.4, we presented the results of two ablation studies. The first study illustrates the impact of the temporal window (recency window) variation which goes both forward and backward in the conversation. For both IEMOCAP sets, the window size of $w = 3$ utterances resulted in the best performance. For MELD dataset a window size of $w = 4$ utterances showed the best results. These results confirm the importance of recency which improves the performance of emotion recognition task in conversations. The second ablation study compares the impact of recency and self-dependency. A complete graph is compared against a conversation graph with recency but without self-dependency and a conversation graph with recency and self-dependency. The performance for IEMOCAP and MELD proved to be the highest when both recency and self-dependency were encoded. These two factors have a significant influence on the performance of the system.

Chapter 5

Conclusion and Future Work

Emotion recognition in conversation is a popular problem, especially in a text-based modality. Despite the significant progress made in text-based emotion detection technology in recent years, speech emotion recognition has not received as much attention. This is even though speech signals are rich in features (e.g., pitch, loudness, and spectral characteristics), that could be used to identify emotions and facilitate the human-computer conversation [13]. Speech emotion recognition in conversation has wide applications including the use in chatbots which has been on the rise in various domains, including education, business, and health. In this thesis, we have adopted a methodology proposed to address SERC by [33]. This methodology is originally proposed for textual data. We adopted the approach to use audio utterances as the input. We used a graph-based approach for utterance level feature extraction and addressed the graph complexity of [33] through graph sparsification. Our model used, a two-stage graph-based approach. In Stage One, audio utterances are input to the model. Each audio utterance is transformed into a cyclic graph follwing a frame-to-node approach. Each speech sample is divided into frames that constitute the nodes in the cyclic-directed graph. The node features are the Low level descriptors (LLDs) which are extracted from the raw audio

utterances using OpenSMILE toolkit. The speech sample graphs are transformed through a two-layer GCN followed by a pooling layer and a fully connected layer to obtain discrete emotion labels. From the pooling layer the utterance-level features are extracted, which is the Stage One output. The features are context-free and speaker-independent.

In Stage Two, a relational conversation graph is constructed. Vertices are initialized using the utterance features extracted in Stage One. Context-sensitive and speaker-sensitive dependencies are modeled in these conversation graphs with relational edges that represent the dependency between the utterances. Using a GNN the context free speaker independant features are transformed into speaker sensitive features. The last stage is a fully connected layer with a softmax layer where the final utterance features are a concatenation of the utterance level features and context and speaker dependant features for emotion classification. The performance of SERC-GCN exceeds existing baselines.

5.1 Summary of contribution

- We developed SERC-GCN (Speech Emotion Recognition in Conversation using GCN), a two-stage graph-based deep learning model for the SERC problem [39]. SERC-GCN predicts a speaker's emotional state by incorporating the *conversational context*, *speaker interactions*, and *utterance temporal dependencies* in the model.

- Our approach to SECR-GCN model is based on adapting and enhancing the approach of the text-based GCN model [33]. The DialogueGCN model [33] utilizes textual data as input and uses a sequential deep learning model to encode the text of each individual utterance. Our approach utilizes audio data and encodes the audio-specific features of each individual utterance. This requires processing and modeling the audio data to extract low level descriptors (LLD's) from raw audio signals. In our work, we have adopted a graph-based approach proposed by [31] for utterance feature extraction and applied to the SERC task. Each audio utterance is transformed into a cyclic graph which follows a frame-to-node

transformation, where M frames (short, overlapping segments) of the speech signal form the M nodes in the directed graph. The stage one architecture has two GCN layers followed by a pooling layer. After the training, utterance-level features are extracted from the pooling layer. These features are the output of Stage One.

- The relational graph-based model follows the approach adopted in the text-based GCN model described in [33], where our model reduces the graph complexity by incorporating graph sparsification. We only consider edges that occurred within a small temporal window to account for the importance of recency of emotions in conversations. During this process a large number of edges are eliminated from the original complete graph while maintaining the most recent ones (depending on the size of the temporal window). We incorporate self-dependancy by preserving a speaker's relation to their own utterances by maintaining all same-speaker edges in the graph. As a result, in this graph-based approach we model both recency and same-spaker dependency for performing SERC.
- For both the IEMOCAP and MELD datasets our SERC-GCN model outperforms standard baselines.
- This work is a step towards incorporating multimodal information for speech emotion recognition in conversation tasks.

From the empirical observations, it can be concluded that the graph convolutional networks for speech emotion recognition in conversations model (SERC-GCN) is an efficient approach and can be used to achieve higher performance for emotion recognition task.

5.2 Future work

In the future, we would like to work on improving the limitations of the model. The following developments can be made on the existing methodology:

- In the future, we would like to explore the task of multi-modal emotion recognition in conversations using a graph-based approach. Incorporating the textual and video modality along with the audio to see how this approach works for multi-modal emotion recognition in conversation [119] and if it can result in a more efficient and accurate recognition system.

- In the current architecture of the model, a softmax layer is used for classification in Stage Two. In the future, we would like to replace it with a more advanced classifier such as Bidirectional Encoder Decoder Transformer (BERT) classifier.

- For Stage One, we can adopt a different strategy. To explore what will happen if the utterance level features are extracted from the fully connected layer instead of the pooling layer.

- In our work, we have illustrated the impact of incorporating context sensitivity, recency, and speaker dependency into the model design for emotion recognition tasks. There are also other factors that exist including the commonsense knowledge, the topic of conversation, and the emotional shift that plays an important role in understanding the complex emotional dynamics. In the future, we can try to explore these factors in the model and see if it can result in more accurate results.

- Recent studies in the direction of ERC have shown that using heterogeneous graphs can improve the performance of the system. As heterogeneity helps to capture semantic information more accurately and improves the node representation [122].

Bibliography

[1] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 527–536.

[2] 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, pp. 335–359, 2008.

[3] C.-C. Hsu, S.-Y. Chen, C.-C. Kuo, T.-H. Huang, and L.-W. Ku, "EmotionLines: An emotion corpus of multi-party conversations," in *International Conference on Language Resources and Evaluation (LREC)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018.

[4] D. Ghosal, N. Majumder, A. Gelbukh, R. Mihalcea, and S. Poria, "COSMIC: CommonSense knowledge for emotion identification in conversations," in *Association for Computational Linguistics: EMNLP*, Online, 2020, pp. 2470–2481.

[5] G. Tu, J. Wen, C. Liu, D. Jiang, and E. Cambria, "Context-and sentiment-aware networks for emotion recognition in conversation," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 699–708, 2022.

[6] L.-N. Do, H.-J. Yang, H.-D. Nguyen, S.-H. Kim, G.-S. Lee, and I.-S. Na, "Deep neural network-based fusion model for emotion recognition using visual data," *The Journal of Supercomputing*, pp. 1–18, 2021.

[7] G. A. R. Kumar, R. K. Kumar, and G. Sanyal, "Facial emotion analysis using deep convolution neural network," in *International Conference on Signal Processing and Communication (ICSPC)*, Malacca, Malaysia, 2017, pp. 369–374.

[8] H. Gunes and M. Piccardi, "Bi-modal emotion recognition from expressive face and body gestures," *Journal of Network and Computer Applications*, vol. 30, no. 4, pp. 1334–1345, 2007.

[9] J. Deng and F. Ren, "A survey of textual emotion recognition and its challenges," *IEEE Transactions on Affective Computing*, 2021.

[10] S. Basu, J. Chakraborty, A. Bag, and M. Aftabuddin, "A review on emotion recognition using speech," in *International conference on inventive communication and computational technologies (ICICCT)*, Coimbatore, India, 2017, pp. 109–114.

[11] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy, "Emotion recognition in conversation: Research challenges, datasets, and recent advances," *IEEE Access*, vol. 7, pp. 100 943–100 953, 2019.

[12] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[13] S. Ramakrishnan and I. M. El Emary, "Speech emotion recognition approaches in human computer interaction," *Telecommunication Systems*, vol. 52, pp. 1467–1478, 2013.

[14] E. Stolterman and A. C. Fors, "Human computer interaction (HCI): Towards a critical research position," *Design Philosophy Papers*, vol. 6, pp. 17–40, 2008.

[15] R. Fleck, "How the move to physical user interfaces can make human computer interaction a more enjoyable experience," in *Physical Interaction (PI03) Workshop on Real World User Interfaces*, Udine, Italy, 2003, p. 71.

[16] B. Shackel, "Human-computer interaction—whence and whither?" *Journal of the American Society for Information Science*, pp. 970–986, 1997.

[17] A. Adem, E. Çakıt, and M. Dağdeviren, "Selection of suitable distance education platforms based on human–computer interaction criteria under fuzzy environment," *Neural Computing and Applications*, vol. 34, no. 10, pp. 7919–7931, 2022.

[18] A. M. Sabelli, T. Kanda, and N. Hagita, "A conversational robot in an elderly care center: an ethnographic study," in *International Conference on Human-Robot Interaction*, Lausanne, Switzerland, 2011, pp. 37–44.

[19] E. Tarawneh, J.-J. Rousseau, S. G. Craig, D. Chandola, W. Khan, A. Faizi, and M. Jenkin, *An Infrastructure for Studying the Role of Sentiment in Human-Robot Interaction.* Cham: Springer Nature Switzerland, 2023, pp. 89–105.

[20] B. Lindblom and J. Sundberg, "The human voice in speech and singing," *Springer Handbook of Acoustics*, pp. 703–746, 2014.

[21] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International Journal of Speech Technology*, vol. 15, pp. 99–117, 2012.

[22] J. Liu, H. Wang, M. Sun, and Y. Wei, "Graph based emotion recognition with attention pooling for variable-length utterances," *Neurocomputing*, vol. 496, pp. 46–55, 2022.

[23] R. Pappagari, P. Żelasko, J. Villalba, L. Moro-Velazquez, and N. Dehak, "Beyond isolated utterances: Conversational emotion recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia, 2021, pp. 39–46.

[24] B. Schuller, A. Batliner, D. Seppi, S. Steidl, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, N. Amir, L. Kessous, and V. Aharonson, "The relevance of feature type for the automatic classification of emotional user states: low level descriptors and functionals," 2007.

[25] R. Jahangir, Y. W. Teh, F. Hanif, and G. Mujtaba, "Deep learning approaches for speech emotion recognition: State of the art and research challenges," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 23 745–23 812, 2021.

[26] Z. Aldeneh and E. M. Provost, "Using regional saliency for speech emotion recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 2741–2745.

[27] J. Huang, Y. Li, J. Tao, Z. Lian *et al.*, "Speech emotion recognition from variable-length inputs with triplet loss function." in *Interspeech*, Hyderabad, India, 2018, pp. 3673–3677.

[28] W. Han, H. Ruan, X. Chen, Z. Wang, H. Li, and B. Schuller, "Towards temporal modelling of categorical speech emotion recognition," in *Interspeech*, Hydrabad, India, 2018, pp. 932–936.

[29] S. Mao, P. Ching, and T. Lee, "Deep learning of segment-level feature representation with multiple instance learning for utterance-level speech emotion recognition." in *Interspeech*, Graz, Austria, 2019, pp. 1686–1690.

[30] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[31] A. Shirian and T. Guha, "Compact graph architecture for speech emotion recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, Ontario, 2021, pp. 6284–6288.

[32] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *International Conference on Neural Information Processing Systems*, ser. NIPS, Red Hook, NY, USA, 2017, p. 1025–1035.

[33] D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, "DialogueGCN: A graph convolutional neural network for emotion recognition in conversation," in *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 154–164.

[34] Z. Lian, J. Tao, B. Liu, J. Huang, Z. Yang, and R. Li, "Conversational emotion recognition using self-attention mechanisms and graph neural networks." in *INTERSPEECH*, Shanghai, China, 2020, pp. 2347–2351.

[35] D. Hazarika, S. Poria, A. Zadeh, E. Cambria, L.-P. Morency, and R. Zimmermann, "Conversational memory network for emotion recognition in dyadic dialogue videos," in *Association for Computational Linguistics. North American Chapter. Meeting*. New Orleans, Louisiana: NIH Public Access, 2018, p. 2122.

[36] C.-W. Huang and S. S. Narayanan, "Attention assisted discovery of sub-utterance structure in speech emotion recognition." pp. 1387–1391, 2016.

[37] N. Majumder, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, and E. Cambria, "Dialoguernn: An attentive rnn for emotion detection in conversations," in *AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, USA, 2019, pp. 6818–6825.

[38] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-vectors meet emotions: A study on dependencies between emotion and speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelone, Spain, 2020, pp. 7169–7173.

[39] D. Chandola, E. Tarawneh, M. Jenkin, and M. Ppapgelis, "SERC-GCN: Speech emotion recognition in conversation using graph convolutional networks," in *IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, Coex, South Korea, 2024.

[40] E. Altarawneh, A. Agrawal, M. Jenkin, and M. Papagelis, "Predicting Evoked Emotions in Conversations," *arXiv e-prints*, p. arXiv:2401.00383, Dec. 2023.

[41] F.-M. Lee, L.-H. Li, and R.-Y. Huang, "Recognizing low/high anger in speech for call centers," in *International Conference on Signal Processing, Robotics and Automation*, Cambridge, UK, 2008, pp. 171–176.

[42] K.-J. Oh, D. Lee, B. Ko, and H.-J. Choi, "A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation," in *IEEE International Conference on Mobile Data Management (MDM)*, Daejeon, South Korea, 2017, pp. 371–375.

[43] D. J. France, R. G. Shiavi, S. Silverman, M. Silverman, and M. Wilkes, "Acoustical properties of speech as indicators of depression and suicidal risk," *IEEE transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 829–837, 2000.

[44] J. H. Hansen and D. A. Cairns, "Icarus: Source generator based real-time recognition of speech in noisy stressful and lombard effect environments," *Speech Communication*, vol. 16, no. 4, pp. 391–422, 1995.

[45] B. Schuller, G. Rigoll, and M. Lang, "Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, Pennsylvania, USA, 2004, pp. I–577.

[46] V. César Cavalcanti Roza and O. Adrian Postolache, "Multimodal approach for emotion recognition based on simulated flight experiments," *Sensors*, vol. 19, no. 24, p. 5516, 2019.

[47] M. S. Hossain, G. Muhammad, B. Song, M. M. Hassan, A. Alelaiwi, and A. Alamri, "Audio–visual emotion-aware cloud gaming framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, pp. 2105–2118, 2015.

[48] R. Nakatsu, J. Nicholson, and N. Tosa, "Emotion recognition and its application to computer agents with spontaneous interactive capabilities," in *ACM International Conference on Multimedia (Part 1)*, Orlando, Florida, USA, 1999, pp. 343–351.

[49] J. Heredia, E. Lopes-Silva, Y. Cardinale, J. Diaz-Amado, I. Dongo, W. Graterol, and A. Aguilera, "Adaptive multimodal emotion detection architecture for social robots," *IEEE Access*, vol. 10, pp. 20 727–20 744, 2022.

[50] D. Lottridge, M. Chignell, and A. Jovicic, "Affective interaction: understanding, evaluating, and designing for human emotion," *Reviews of Human Factors and Ergonomics*, vol. 7, no. 1, pp. 197–217, 2011.

[51] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[52] K. T. Strongman, *The Psychology of Emotion: Theories of Emotion in Perspective*. New York: John Wiley & Sons, 1996.

[53] H.-M. Gardiner, R. C. Metcalf, and J. G. Beebe-Center, "Feeling and emotion: A history of theories." 1937.

[54] A. Moors, "Theories of emotion causation: A review," *Cognition and Emotion*, vol. 23, no. 4, pp. 625–662, 2009.

[55] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[56] M. A. Mohsin and A. Beltiukov, "Summarizing emotions from text using plutchik's wheel of emotions," in *Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS)*, Ufa, Russia., 2019, pp. 291–294.

[57] F. A. Acheampong, C. Wenyu, and H. Nunoo-Mensah, "Text-based emotion detection: Advances, challenges, and opportunities," *Engineering Reports*, vol. 2, no. 7, p. e12189, 2020.

[58] S. Y. M. Lee, Y. Chen, C.-R. Huang, and S. Li, "Detecting emotion causes with a linguistic rule-based approach 1," *Computational Intelligence*, vol. 29, no. 3, pp. 390–416, 2013.

[59] V. Hozjan and Z. Kačič, "A rule-based emotion-dependent feature extraction method for emotion analysis from speech," *The Journal of the Acoustical Society of America*, vol. 119, no. 5, pp. 3109–3120, 2006.

[60] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Deep learning-based sentiment classification of evaluative text based on multi-feature fusion," *Information Processing & Management*, vol. 56, no. 4, pp. 1245–1259, 2019.

[61] D. Xu, Z. Tian, R. Lai, X. Kong, Z. Tan, and W. Shi, "Deep learning based emotion analysis of microblog texts," *Information Fusion*, vol. 64, pp. 1–11, 2020.

[62] U. Rashid, M. W. Iqbal, M. A. Skiandar, M. Q. Raiz, M. R. Naqvi, and S. K. Shahzad, "Emotion detection of contextual text using deep learning," in *International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. Istanbul,Turkey: IEEE, 2020, pp. 1–5.

[63] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of bert-based approaches," *Artificial Intelligence Review*, pp. 1–41, 2021.

[64] M.-H. Su, C.-H. Wu, K.-Y. Huang, and Q.-B. Hong, "Lstm-based text emotion recognition using semantic and emotional word vectors," in *Asian Conference on Affective Computing and Intelligent Interaction (ACII Asia)*, San Antonio, TX, USA, 2018, pp. 1–6.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[66] L. Luo and Y. Wang, "Emotionx-hsu: Adopting pre-trained BERT for emotion classification," *CoRR*, vol. abs/1907.09669, 2019.

[67] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A dataset of fine-grained emotions," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4040–4054.

[68] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:52967399

[69] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[70] A. F. Adoma, N.-M. Henry, and W. Chen, "Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition," in *International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China., 2020, pp. 117–121.

[71] Y. Chavhan, M. Dhore, and P. Yesaware, "Speech emotion recognition using support vector machines," *International Journal of Computer Applications*, vol. 1, no. 20, pp. 6–9, 2010.

[72] T. Vogt and E. Andre, "Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition," in *IEEE International Conference on Multimedia and Expo*, Amsterdam, The Netherlands, 2005, pp. 474–477.

[73] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011," *Artificial Intelligence Review*, vol. 43, 02 2012.

[74] J. R. Deller Jr, "Discrete-time processing of speech signals," in *Discrete-time processing of speech signals*. Prentice Hall PTR, 1993, pp. 908–908.

[75] B. Schuller, R. Müller, M. Lang, and G. Rigoll, "Speaker independent emotion recognition by early fusion of acoustic and linguistic features within ensemble," in *Europ. Conf. on Speech Communication and Technology, Interspeech*, Lisbon, Portugal, 2005.

[76] Y. Wang and L. Guan, "An investigation of speech-based human emotion recognition," in *IEEE Workshop on Multimedia Signal Processing*, 2004, pp. 15–18.

[77] H. K. Mishra and C. C. Sekhar, "Variational gaussian mixture models for speech emotion recognition," in *International Conference on Advances in Pattern Recognition (ICPR)*, Kolkata, India, 2009, pp. 183–186.

[78] A. D. Dileep and C. C. Sekhar, "Gmm-based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1421–1432, 2014.

[79] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Networks*, vol. 92, pp. 60–68, 2017.

[80] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[81] Y. Bengio *et al.*, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[82] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[83] Z. Cairong, Z. Xinran, Z. Cheng, and Z. Li, "A novel dbn feature fusion model for cross-corpus speech emotion recognition," *Journal of Electrical and Computer Engineering*, 2016.

[84] R. D. Fonnegra and G. M. Díaz, "Speech emotion recognition based on a recurrent neural network classification model," in *International Conference on Advances in Computer Entertainment*. London, UK: Springer, 2017, pp. 882–892.

[85] M. N. Stolar, M. Lech, R. S. Bolia, and M. Skinner, "Real time speech emotion recognition using rgb image classification and transfer learning," in *International Conference on Signal Processing and Communication Systems (ICSPCS)*, Surfers Paradise, Australia, 2017, pp. 1–8.

[86] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[87] N. E. Cibau, E. M. Albornoz, and H. L. Rufiner, "Speech emotion recognition using a deep autoencoder," *Anales de la XV Reunion de Procesamiento de la Informacion y Control*, vol. 16, pp. 934–939, 2013.

[88] Y. Xie, R. Liang, Z. Liang, and L. Zhao, "Attention-based dense lstm for speech emotion recognition," *IEICE Transactions on Information and Systems*, vol. 102, no. 7, pp. 1426–1429, 2019.

[89] J. Wang, M. Xue, R. Culhane, E. Diao, J. Ding, and V. Tarokh, "Speech emotion recognition with dual-sequence lstm architecture," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 6474–6478.

[90] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1d & 2d cnn lstm networks," *Biomedical signal processing and control*, vol. 47, pp. 312–323, 2019.

[91] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, "Speech emotion recognition using cnn," 11 2014, pp. 801–804.

[92] W. Zheng, J. Yu, and Y. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *International Conference on Affective Computing and Intelligent Interaction (ACII)*. Xi'an, China: IEEE, 2015, pp. 827–831.

[93] L. Sun, J. Chen, K. Xie, and T. Gu, "Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition," *International Journal of Speech Technology*, vol. 21, pp. 931–940, 2018.

[94] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1–9.

[95] M. Stolar, M. Lech, R. Bolia, and M. Skinner, "Real time speech emotion recognition using rgb image classification and transfer learning," 12 2017, pp. 1–8.

[96] A. Badshah, J. Ahmad, N. Rahim, and S. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," 02 2017, pp. 1–5.

[97] A. Tursunov, M. Khan, and S. Kwon, "Deep-net: A lightweight cnn-based speech emotion recognition system using deep frequency features," *Sensors*, vol. 20, p. 5212, 09 2020.

[98] R. Li, Z. Wu, X. Liu, H. Meng, and L. Cai, "Multi-task learning of structured output layer bidirectional lstms for speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, 2017, pp. 5510–5514.

[99] I. Shahin, N. Al Hindawi, A. Nassif, A. Alhudhaif, and K. Polat, "Novel dual-channel long short-term memory compressed capsule networks for emotion recognition," *Expert Systems with Applications*, vol. 188, 10 2021.

[100] R. Nagase, T. Fukumori, and Y. Yamashita, "Speech emotion recognition using label smoothing based on neutral and anger characteristics," in *Global Conference on Life Sciences and Technologies (LifeTech)*, Osaka, Japan, 2022, pp. 626–627.

[101] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[102] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, Toulon, France, 2017.

[103] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[104] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[105] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[106] M. K. Islam, S. Aridhi, and M. Smail-Tabbone, "A comparative study of similarity-based and gnn-based link prediction approaches," *arXiv preprint arXiv:2008.08879*, 2020.

[107] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[108] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[109] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[110] L. Gutiérrez-Gómez and J.-C. Delvenne, "Unsupervised network embeddings with node identity awareness," *Applied Network Science*, vol. 4, no. 1, pp. 1–21, 2019.

[111] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, p. 3844–3852, 2016.

[112] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[113] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations (ICLR)*, New Orleans, 2019.

[114] L. Wei, H. Zhao, Q. Yao, and Z. He, "Pooling architecture search for graph classification," in *ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2021, p. 2091–2100.

[115] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," vol. 97. Proceedings of Machine Learning Research, 2019, pp. 3744–3753.

[116] Z. Pan, Z. Luo, J. Yang, and H. Li, "Multi-modal attention for speech emotion recognition," *arXiv preprint arXiv:2009.04107*, 2020.

[117] D. Krishna and A. Patil, "Multimodal emotion recognition using cross-modal attention and 1d convolutional neural networks." in *Interspeech*, Shanghai, China, 2020, pp. 4243–4247.

[118] J.-H. Lee, H.-J. Kim, and Y.-G. Cheong, "A multi-modal approach for emotion recognition of tv drama characters using image and text," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea (South), 2020, pp. 420–424.

[119] J. Li, X. Wang, G. Lv, and Z. Zeng, "Graphmft: A graph network based multimodal fusion technique for emotion recognition in conversation," *Neurocomputing*, vol. 550, p. 126427, 2023.

[120] D. Zhang, L. Wu, C. Sun, S. Li, Q. Zhu, and G. Zhou, "Modeling both context-and speaker-sensitive dependence for emotion detection in multi-speaker conversations." in *IJCAI*, 2019, pp. 5415–5421.

[121] J. Hu, Y. Liu, J. Zhao, and Q. Jin, "Mmgcn: Multimodal fusion via deep graph convolution network for emotion recognition in conversation," *ArXiv*, vol. abs/2107.06779, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235829068

[122] Y. Liang, F. Meng, Y. Zhang, Y. Chen, J. Xu, and J. Zhou, "Emotional conversation generation with heterogeneous graph neural network," *Artificial Intelligence*, vol. 308, p. 103714, 2022.

[123] D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*, vol. 48, pp. 1162–1181, 09 2006.

[124] M. Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, pp. 572–587, 2011.

[125] M. C. Sezgin, B. Gunsel, and G. K. Kurt, "Perceptual audio features for emotion detection," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2012, no. 1, pp. 1–21, 2012.

[126] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *ACM International Conference on Multimedia*, New York, United States, 2010, pp. 1459–1462.

[127] F. Eyben, M. Wöllmer, and B. Schuller, "Openear: Introducing the munich open-source emotion and affect recognition toolkit," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, 2009, pp. 1–6.

[128] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, no. CONF, Waikoloa, HI, USA,, 2011.

[129] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, "Covarep—a collaborative voice analysis repository for speech technologies," in *International Conference on Acoustics, Speech and Signal Processing ICASSP*, Florence, Italy, 2014, pp. 960–964.

[130] T. Giannakopoulos, "pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, p. e0144610, 2015.

[131] J. Lyons, "Python speech features," *Dostupné z: https://github. com/jameslyons/python speech features*, p. 2018, 2013.

[132] [Online]. Available: https://www.audeering.com/research/opensmile/

[133] X. Ma, Z. Wu, J. Jia, M. Xu, H. Meng, and L. Cai, "Speech emotion recognition with emotion-pair based framework considering emotion distribution information in dimensional emotion space." in *Interspeech*, 2017, pp. 1238–1242.

[134] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Stockholm, Sweden, 2017, pp. 2227–2231.

[135] D. Luo, Y. Zou, and D. Huang, "Investigation on joint representation learning for robust feature extraction in speech emotion recognition." in *Interspeech*, Hyderabad, India, 2018, pp. 152–156.

[136] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and J. Epps, "Direct modelling of speech emotion from raw speech," *arXiv preprint arXiv:1904.03833*, 2019.

[137] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017*.

[138] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *CoRR*, vol. abs/1312.6203, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:17682909

[139] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 2014–2023.

[140] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *International Conference on Neural Information Processing Systems*, ser. NIPS. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 4805–4815.

[141] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[142] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio', and Y. Bengio, "Graph attention networks," *ArXiv*, vol. abs/1710.10903, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3292002

[143] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *International Conference on Neural Information Processing Systems*, ser. NIPS, Red Hook, NY, USA, 2017, p. 1025–1035.

[144] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 3744–3753.

[145] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency, "Context-dependent sentiment analysis in user-generated videos," in *Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 873–883.

[146] D. Hazarika, S. Poria, R. Mihalcea, E. Cambria, and R. Zimmermann, "Icon: Interactive conversational memory network for multimodal emotion detection," in *Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2594–2604.

[147] [Online]. Available: https://en.wikipedia.org/wiki/Friends

[148] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria, "A survey on empathetic dialogue systems," *Information Fusion*, vol. 64, pp. 50–70, 2020.

[149] Z. Lin, A. Madotto, J. Shin, P. Xu, and P. Fung, "MoEL: Mixture of empathetic listeners," in *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 121–132.

[150] N. Majumder, P. Hong, S. Peng, J. Lu, D. Ghosal, A. Gelbukh, R. Mihalcea, and S. Poria, "MIME: MIMicking emotions for empathetic response generation," in *Empirical Methods in Natural Language Processing (EMNLP)*. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2020, pp. 8968–8979.

[151] Q. Li, H. Chen, Z. Ren, P. Ren, Z. Tu, and Z. Chen, "EmpDG: Multi-resolution interactive empathetic dialogue generation," in *International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 4454–4466.

[152] R. Zhang, Z. Wang, and D. Mai, "Building emotional conversation systems using multi-task seq2seq learning," in *National CCF Conference on Natural Language Processing and Chinese Computing (NLPCC)*. Dalian, China: Springer, 2018, pp. 612–621.

[153] N. Asghar, I. Kobyzev, J. Hoey, P. Poupart, and M. B. Sheikh, "Generating emotionally aligned responses in dialogues using affect control theory," *arXiv preprint arXiv:2003.03645*, 2020.

[154] L. Chen, X. Mao, Y. Xue, and L. L. Cheng, "Speech emotion recognition: Features and classification models," *Digital signal processing*, vol. 22, no. 6, pp. 1154–1160, 2012.

[155] C. Shan, "Learning human emotion from body gesture," *Encyclopedia of the Sciences of Learning*, pp. 1887–1889, 2012.

[156] N. Lubis, S. Sakti, K. Yoshino, and S. Nakamura, "Eliciting positive emotion through affect-sensitive dialogue response generation: A neural network approach," in *AAAI conference on artificial intelligence*, vol. 32, no. 1, New Orleans, Louisiana, USA, 2018.

[157] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *AAAI Conference on Artificial Intelligence*, no. 1, San Francisco, California, 2017.

[158] B. L. Fredrickson, "The role of positive emotions in positive psychology: The broaden-and-build theory of positive emotions." *American psychologist*, vol. 56, no. 3, p. 218, 2001.

[159] B. Koptyra, A. Ngo, Ł. Radliński, and J. Kocoń, "Clarin-emo: Training emotion recognition models using human annotation and chatgpt," in *Computational Science – ICCS*. Cham: Springer Nature Switzerland, 2023, pp. 365–379.

[160] V. Sorin, D. Brin, Y. Barash, E. Konen, A. Charney, G. Nadkarni, and E. Klang, "Large language models (llms) and empathy-a systematic review," *medRxiv*, pp. 2023–08, 2023.

[161] S. Tripathi and H. S. M. Beigi, "Multi-modal emotion recognition on IEMOCAP dataset using deep learning," *CoRR*, vol. abs/1804.05788, 2018. [Online]. Available: http://arxiv.org/abs/1804.05788

[162] W. Han, H. Ruan, X. Chen, Z. Wang, H. Li, and B. Schuller, "Towards Temporal Modelling of Categorical Speech Emotion Recognition," in *Proc. INTERSPEECH*, India, 2018, pp. 932–936.

[163] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, "Convolutional MKL based multimodal emotion recognition and sentiment analysis," in *International Conference on Data Mining (ICDM)*, Barcelona, 2016, pp. 439–448.

[164] M. Neumann and N. T. Vu, "Improving speech emotion recognition with unsupervised representation learning on unlabeled speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 7390–7394.

[165] [Online]. Available: https://pytorch-geometric.readthedocs.io/en/latest/