



Titre: Multimodal Emotion Recognition and Sentiment Analysis
Title:

Auteur: Rihab Hajlaoui
Author:

Date: 2023

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Hajlaoui, R. (2023). Multimodal Emotion Recognition and Sentiment Analysis
Citation: [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
<https://publications.polymtl.ca/57077/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/57077/>
PolyPublie URL:

**Directeurs de
recherche:** Guillaume-Alexandre Bilodeau
Advisors:

Programme: Génie informatique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Multimodal Emotion Recognition and Sentiment Analysis

RIHAB HAJLAOUI

Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie informatique

Décembre 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Multimodal Emotion Recognition and Sentiment Analysis

présenté par **Rihab HAJLAOUI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Michel DESMARAIS, président

Guillaume-Alexandre BILODEAU, membre et directeur de recherche

Quentin CAPPART, membre

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to those who provided me with the possibility of accomplishing this project. Special gratitude to give to my supervisor **Dr. Guillaume-Alexandre Bilodeau**, because of his suggestions and encouragement which helped me to achieve the project's realization. I would like to thank also **M. Jan Rockemann**, for his advice and help throughout the internship. Finally, a special thanks to my family for their support and help when it was most needed.

RÉSUMÉ

Le développement de l’intelligence artificielle et de ses sous-domaines tels que la vision artificielle et le traitement du langage naturel au cours de la dernière décennie a contribué à une progression majeure dans les domaines de la reconnaissance des émotions et de l’analyse des sentiments.

Depuis plusieurs décennies, les chercheurs se concentrent sur la construction de modèles d’apprentissage profond capables de reconnaître l’état émotionnel d’une personne en analysant et en explorant les caractéristiques d’une seule modalité telle que le texte, la parole ou l’image.

Bien que les performances des systèmes unimodaux ont beaucoup progressées, l’utilisation d’une seule modalité à la fois a pour conséquence d’ignorer des informations importantes qui pourraient exister dans d’autres modalités. Il est donc essentiel de développer un système de reconnaissance multimodale des émotions capable d’explorer différentes modalités de données pour prédire efficacement les émotions et les sentiments.

Dans ce travail, nous proposons un nouveau réseau transformateur multimodal en intégrant les réseaux résiduels à portes pour améliorer les interactions entre les différentes modalités de données.

En outre, nous étudions la capacité des transformateurs pré-entraînés sur des tâches de langage naturel à améliorer les performances de la tâche de reconnaissance des émotions multimodales en aval. Plusieurs architectures ont été proposées pour affiner le modèle GPT-2 pré-entraîné afin d’intégrer les modalités image, texte et audio, et d’apprendre les différentes interactions entre ces modalités.

Nous avons évalué l’efficacité des modèles proposés sur trois ensembles de données de référence : CMU-MOSI, CMU-MOSEI et IEMOCAP. En utilisant un transformateur multimodal avec GRN, nous avons obtenu des résultats compétitifs qui ont surpassé les performances de plusieurs modèles de pointe dans trois ensembles de données de référence. En outre, parmi les modèles pré-entraînés basés sur des transformateurs, certaines architectures proposées ont obtenu des résultats comparables, en particulier sur l’ensemble de données IEMOCAP.

ABSTRACT

The rise of artificial intelligence and its sub-fields such as computer vision and natural language processing in the last decade has had a major contribution to the progression of the fields of emotion recognition and sentiment analysis.

For many decades, researchers have been focusing on building deep-learning models that are able to recognize a person’s emotional state by analyzing and exploring the features of a single modality such as text, speech, or image.

Although unimodal systems have steadily progressed in performance, using only a single modality at the time has the consequence of omitting important information that could exist in other modalities. Therefore, it is crucial to develop a multimodal emotion recognition system that is able to explore different modalities of data to effectively predict emotions and sentiments.

In this work, we propose a new multimodal transformer network by integrating Gated Residual Networks to improve the interactions between the different modalities of data.

Furthermore, we investigated the capability of transformers that are pre-trained on natural language tasks to improve the performance of the downstream multimodal emotion recognition task. Several architectures have been proposed to finetune a pre-trained GPT-2 model to embed image, text, and audio modalities, and to learn the different interactions between these modalities.

We evaluated the effectiveness of our proposed models on three benchmark datasets CMU-MOSI, CMU-MOSEI, and IEMOCAP. Employing a multimodal transformer with GRN, we attained competitive results that surpassed the performance of several state-of-the-art models across three benchmark datasets. Additionally, among the pre-trained transformer-based models, certain proposed architectures demonstrated comparable results specifically on the IEMOCAP dataset.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF SYMBOLS AND ACRONYMS	xv
CHAPTER 1 INTRODUCTION	1
1.1 Objectives	4
1.2 Thesis outline	4
CHAPTER 2 BACKGROUND AND RELATED WORKS	6
2.1 Machine learning background	6
2.1.1 Supervised Learning	6
2.2 Deep learning models	7
2.2.1 Perceptron	7
2.2.2 Convolutional Neural Network	9
2.2.3 Recurrent Neural Networks (RNNs)	11
2.2.4 Long-Short-Term-Memory (LSTM)	13
2.2.5 Transformers	14
2.2.6 Gated Residual Networks	15
2.3 Losses and hyperparameters	17
2.4 Related work	19
2.4.1 Text-based Methods	19
2.4.2 Image-based Methods	22
2.4.3 Audio-based Methods	24
2.4.4 Multimodal-based Methods	25
2.5 Discussion	27

CHAPTER 3	OVERVIEW OF THE METHODS	28
3.1	Research Methodology	28
3.2	Dataset	29
3.3	Software and Hardware Environments	30
3.4	Contributions	30
CHAPTER 4	ARTICLE 1: MTGR: IMPROVING EMOTION AND SENTIMENT ANALYSIS WITH GATED RESIDUAL NETWORKS	31
4.1	Introduction	32
4.2	Related Work	33
4.3	Methodology	34
4.3.1	Problem Definition	34
4.3.2	Details of the proposed method	35
4.4	Experiments	38
4.4.1	Datasets and evaluation metrics	38
4.4.2	Implementation details	39
4.4.3	Results and discussion	40
4.5	Conclusion	41
CHAPTER 5	MULTIMODAL PRE-TRAINED TRANSFORMERS FOR EMOTION RECOGNITION AND SENTIMENT ANALYSIS	42
5.1	Pre-trained GPT-2 transformer model	43
5.2	Fusion mechanisms	43
5.2.1	Architecture A: Concatenation	45
5.2.2	Architecture B: GRN and high-level features	45
5.2.3	Architecture C: GRN and low-level features	46
5.2.4	Architecture D: Co-attention and low-level features	48
5.2.5	Architecture E: Co-attention for low-level features and GRN for high-level features	49
5.2.6	Architecture F: Co-attention and GRN for high-level features	50
5.3	Fine-tuning	51
5.4	Experiments	52
5.4.1	Comparison of the proposed architectures	52
5.4.2	Comparison with the state-of-the-art	54
CHAPTER 6	ANALYSIS AND DISCUSSION	56
6.1	Custom transformer vs GPT-2	56

6.2 Analysis of the GPT-2-based models	56
CHAPTER 7 CONCLUSION	59
7.1 Summary of Works	59
7.2 Future Research	59
REFERENCES	60

LIST OF TABLES

Table 4.1	Hyperparameters	39
Table 4.2	Results on the CMU-MOSEI Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	40
Table 4.3	Results on the CMU-MOSI Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	41
Table 4.4	Results on the IEMOCAP Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	41
Table 5.1	Architectures/models and their modality inputs	52
Table 5.2	Results on the CMU-MOSEI Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	53
Table 5.3	Results on the CMU-MOSI Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	53
Table 5.4	Results on the IEMOCAP Dataset. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	54
Table 5.5	Results on CMU-MOSEI Dataset in Comparison to State-of-the-Art Models. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	55

Table 5.6	Results on CMU-MOSI Dataset in Comparison to State-of-the-Art Models. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	55
Table 5.7	Results on IEMOCAP Dataset in Comparison to State-of-the-Art Models. Boldface : best results. ↑: the higher value of the metrics means the better performance of the model. ↓: the lower value of the metrics means the better performance of the model.	55

LIST OF FIGURES

Figure 1.1	Multimodal Emotion Recognition/Sentiment Analysis System	3
Figure 2.1	Within a Perceptron, each input is multiplied by an associated weight. The resulting weighted inputs are aggregated and processed by an activation function, determining the output. The goal is to seek the ideal weight values that minimize the error [1] © Roi Yehoshua, 2023. . . .	8
Figure 2.2	A standard convolutional neural network (CNN) architecture comprises feature learning layers, typically consisting of a combination of pooling and convolution operations applied to the input image, as well as a classification layer, typically implemented as a fully connected layer [2] © Sumit Saha, 2018.	9
Figure 2.3	In a convolution operation, a kernel, also referred to as a filter, slides over the entire input image region by region. Within each region, the values of the input image and the kernel are subjected to element-wise multiplication. The outcomes of these multiplications are aggregated by summation [3] © Copyright 2021, Janosh Riebesell.	10
Figure 2.4	During the pooling operation, a window slides over the input image region by region. In the case of max pooling, the highest value within the window is selected at each region, while average pooling involves computing the mean value of the window elements [4] © 2018, David S. Batista.	10
Figure 2.5	The RNN's output at each time step t is determined by the current input X_t and the hidden state from the preceding time step S_{t-1} . [5] © Aman Singh, 2021.	12
Figure 2.6	From left to right: "One to One" RNN architecture involves a single input connected to a single output. "One to Many" occurs when a single input generates multiple outputs. "Many to One" denotes a scenario where the RNN takes a sequence of inputs and predicts a single output. "Many to Many" when the RNN predicting a sequence of outputs based on a sequence of inputs. "Many to Many (2)" refers to a specific case in which each element in the input sequence of the RNN corresponds directly to a specific element in the output sequence [6] © Isheunesu Tembo, 2019.	12

- Figure 2.7 At each time step t , the LSTM cell takes the current input X_t the preceding cell memory h_{t-1} , and the previous cell state C_{t-1} as inputs. The input gate f_t applies a sigmoid function to a combination of the current input X_t and the previous memory state h_{t-1} . Similarly, the input gate i_t also employs a sigmoid function on a combination of the current input and the previous memory state. The new input value $\hat{C}_t$ is obtained as the $\tanh$ of a linear combination of the current input X_t and the previous memory state h_{t-1} . The cell state C_t is then computed based on the input gate i_t , the forget gate f_t , and the new input value $\hat{C}_t$. Finally, the output O_t is calculated based on the current input X_t and the previous memory state h_{t-1} . [7] © Thorir Mar Ingolfsson, 2021. 14
- Figure 2.8 (1) On the left side, we observe the computations taking place within the scaled dot-product attention mechanism. This operation takes key (K), query (Q), and value (V) vectors as input. Initially, it computes the dot product between each query and all key vectors, resulting in attention scores. Then, these attention scores go through a scaling process, followed by the application of a softmax function to generate attention weights for each key vector. Finally, these weights are employed to calculate the weighted sum of the value vectors. (2) On the right side, we have the concept of multi-head attention, which involves the parallel calculation of attention scores using multiple attention heads. [8] © Vaclav Kosar, 2022. 15
- Figure 2.9 A Gated Residual Network layer takes two inputs: X as the primary input and Y as an optional secondary input. It includes skip connections and gating layers capable of dynamically learning how to combine the two inputs. 16
- Figure 2.10 "Too low": When the learning rate is set too low, a large number of weight updates is required to minimize the error function, which results in slow convergence. "Just right": The learning rate is appropriately set. "Too high ": If the learning rate is too high, the model parameters are updated with large steps, preventing proper convergence of the model [9] © Jeremy Jordan, 2018. 18

Figure 4.1	Overview of the method. The main components of the multimodal transformer are the crossmodal transformers which capture the dynamics between two modalities of data. The outputs of the crossmodal transformers are concatenated and passed through a self-attention transformer to capture temporal information for the prediction.	34
Figure 4.2	Crossmodal Transformer Block between two modalities α and β , where α is the target modality and β is the source modality. The main components of the crossmodal transformer are the co-attention operation to calculate the enhanced features for the target modality, and the Gated Residual Network to suppress unnecessary information coming from the source modality.	35
Figure 4.3	Gated Residual Network	37
Figure 5.1	GPT-2 architecture is mainly composed of positional embeddings combined with token embeddings, and decoder blocks. Each decoder block comprises a masked self-attention layer and a feedforward layer [10] © Cameron R. Wolfe, 2022.	43
Figure 5.2	Overview of multimodal pre-trained transformer architectures is presented here. The initial architecture involves the fusion of low-level features extracted from diverse data modalities. The resultant features are subsequently inputted into a pre-trained transformer, such as GPT-2, to extract high-level features. These high-level features are then employed by a feedforward network to predict emotions. In contrast, the second architecture takes a different approach. It first extracts high-level features independently from each modality using GPT-2 transformers. Subsequently, the outputs of these GPT-2 models are fused, and the resulting feature vector is fed into a feedforward network to predict emotions.	44
Figure 5.3	The architecture featuring GPT-2 with concatenation as the fusion method. The initial step involves the concatenation of the modalities X_1 to X_n , forming a combined input, which is then fed into a pre-trained GPT-2 model for the extraction of high-level features. Subsequently, the resulting vector is passed into a feedforward network for making predictions.	45

Figure 5.4	The architecture featuring GPT-2 with GRN serving as the fusion method for high-level features. The process begins by extracting high-level features utilizing pre-trained GPT-2 models. Following this, GRN is employed to combine the outputs from the GPT-2 models. Finally, the resultant vector is fed into a feedforward network to make predictions.	46
Figure 5.5	The architecture featuring GPT-2 with GRN serving as the fusion method for low-level features. The process begins by using GRNs to amalgamate the projected input modalities. The resulting fused vector is subsequently input into a pre-trained GPT-2 to extract a high-level feature vector, which is then utilized by a feedforward network for predicting results.	47
Figure 5.6	The architecture featuring GPT-2 with co-attention layers as the fusion method for low-level features. The process begins by using co-attention layers to amalgamate the projected input modalities. The resulting fused vector is subsequently input into a pre-trained GPT-2 to extract a high-level feature vector, which is then utilized by a feedforward network for predicting results.	49
Figure 5.7	The architecture featuring GPT-2 with co-attention layers as the fusion method for low-level features and GRN for high-level features. The process begins by using co-attention layers to merge two distinct modalities. The resultant enhanced feature vectors are then input into a pre-trained GPT-2 to derive high-level feature vectors. Following this, GRN is employed to combine the high-level features. Finally, the resulting vector from GRN is forwarded into a feedforward network for prediction.	50
Figure 5.8	The architecture featuring GPT-2 along with co-attention layers and GRNs for fusing high-level features. Initially, GPT-2 extracts high-level feature vectors. Subsequently, co-attention layers enhance these high-level feature vectors, and the improved features are combined using a GRN layer. Following this, the resulting vector from the GRN is fed into a feedforward network for making predictions.	51

LIST OF SYMBOLS AND ACRONYMS

DL	Deep Learning
MuT	Multimodal Transformer
GRN	Gated Residual Network
MTGR	Multimodal Transformer with Gated Residual Networks
ML	Machine Learning
CNN	Convolutional Neural Network
ConvNet	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long-Short-Term-Memory
CV	Computer Vision
NLP	Natural Language Processing
KBANN	knowledge-based Artificial Neural Network
SVM	Support Vector Machine
LR	Logistic Regression
BiLSTM	Bidirectional long short-term memory
BERT	Bidirectional Encoder Representations from Transformers
RPCA	Robust Principal Component Analysis
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
KNN	K-nearest neighbor algorithm
LBP	Local Binary Pattern
MANFIS	Multiple Adaptive Neuro-Fuzzy Networks
LPTP	Local Principal Texture Pattern
VGG	Visual Geometry Group
ResNet	Residual Neural Network
ViT	Visual Transformer
DAN	Distract your Attention Network
MAN	Multi-head Attention Network
AFN	Attention Fusion Network
APP	Attentive Patch Pooling
ATP	Attentive Token Pooling
GMM	Gaussian Mixture Model
MLP	Multi-Layer Perceptron

MDT	Meta Decision Tree
MaxEnt	Maximum Entropy Model
EARs	Emotion Association Rules
CRNN	Convolutional Recurrent Neural Network
ANN	Artificial Neural Network
TFN	Tensor Fusion Network
Bi-GRU	Bidirectional Gated Recurrent Unit
RAVEN	Recurrent Attended Variation Embedding Network
GLU	Gated Linear Unit
ELU	Exponential Linear Unit
GPT-2	Generative Pre-trained Transformer 2
FPT	Frozen Pre-trained Transformer
CLIP	Contrastive Language Image Pretraining

CHAPTER 1 INTRODUCTION

Emotions can be defined as reactions to an event that can manifest themselves in physiological, cognitive, and behavioural ways in an individual. Emotions play an essential role in the life of a human being. Whatever the nature of the emotion (joy, sadness, anger, etc.) or the intensity of the emotion (strong, weak), all types of emotions are necessary to protect a person's integrity and well-being. They influence perception, attention, learning, memory, reasoning, and decision handling.

Therefore, emotion recognition and sentiment analysis are essential to achieve natural human-computer interaction. They have attracted the attention of many researchers and developers due to the need to interpret human behaviour in many fields. For instance, it can be used as a diagnostic tool for therapists to effectively treat and identify clinical conditions, such as neurodevelopmental, neurological, and psychiatric disorders [11]. It can also be used in the intelligent vehicle industry, where an onboard system can be developed to predict and provide the driver's mental state, which can improve safety and prevent potential accidents [12]. In addition, emotion recognition can also be useful in speech detection or machine translation, where the person's emotional state can provide a more accurate translation [13] [14]. In the marketing sector, the analysis of customers' emotional reactions or sentiments can dynamically improve the activities of companies or agencies by enhancing the customer experience and providing better products or services [15] [16]. Finally, emotion recognition can further be used in video games, call centers, education, and more.

In our case, we consider the use of emotion detection to offer an automated decision support system for human resource (HR) management. With the use of information provided by human experts, legal, policy, and employment documents (various sources and modalities of data), the objective is to develop an AI-based model that can recognize emotions and analyze sentiments. Therefore, the model will help to recommend the appropriate course of action for the HR department to take when dealing with employees.

Recognizing human emotions and feelings is a very challenging task, as emotions can be expressed differently from one person to another. Depending on many factors (personality, culture, environment, gender, etc.), a person may perceive and interact with his or her emotions differently. Often, emotional reactions to a particular event can be conveyed verbally using verbal communication through spoken or written words, or they can also be conveyed using non-verbal communication cues, such as facial expressions, gestures, body language, and so on.

At present, most research has focused on unimodal systems, notably using textual data given that people interact with each other primarily by means of words on online platforms. Text data has been used to a greater extent because, unlike other forms of communication, language can be considered the most effective and unambiguous way to express human emotions and feelings [17]. An individual’s facial expressions, body gestures, or tone of voice may be misleading or insufficient in expressing their current emotional state. In addition, when it comes to research involving data analysis, the processing of visual data is far more complex than that of textual data. The process of analyzing visual data involves many complex steps, such as detecting individuals in images, processing facial images, extracting distinctive features, and, finally, identifying the emotions expressed. The complexity of visual data poses a range of challenges for AI-based models. For example, the model needs to be trained using a variety of images, encompassing different angles, lighting conditions, and backgrounds. In addition, the dataset must involve a wide range of individuals, including people of different genders, nationalities, cultures, and races, to ensure the model effectiveness and accuracy in detecting emotions.

Many researchers have used Deep Learning (DL) and its related techniques to develop systems to analyze sentiments or identify human feelings based on textual input. Nevertheless, other modalities were also considered. Researchers have also proposed models for automatically recognizing emotions or sentiments from facial images using facial actions; others have relied on body movements. Furthermore, some other studies have been conducted in order to recognize human vocal emotions using features extracted from speech.

Although unimodal emotion recognition and sentiment analysis systems have had great success, relying solely on one single cue or one single modality of data is not sufficient to develop an effective automatic emotion recognition or sentiment analysis system. There are some instances in which not all modalities can convey an individual’s emotional state. For example, a neutral speech could be accompanied by a sad facial expression. In other cases, the different modalities may contradict each other. For example, a feeling of sadness or unease may be accompanied by a fake smile.

In order to make effective progress in the field of emotional intelligence and develop a more robust automatic emotion recognition system, a machine needs to have access to different forms of communication (verbal and non-verbal), as in the case of an interaction between two people. Therefore, most recent research has focused on developing multimodal emotion recognition systems that use different modalities of data (text, image, and audio). As shown in Figure 1.1, a multimodal emotion recognition system will combine different modalities of data to detect emotions or sentiments.

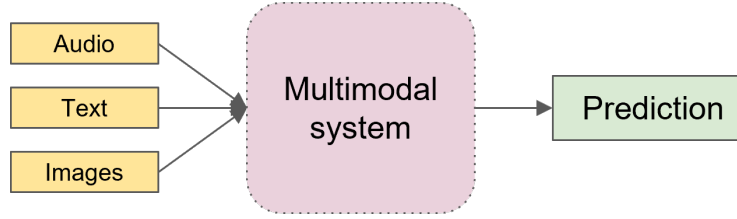


Figure 1.1 Multimodal Emotion Recognition/Sentiment Analysis System

Although the use of multiple data modalities offers significant advantages and additional information, multimodal emotion recognition represents a significant challenge due to several key factors. Firstly, the efficient integration of heterogeneous features from different data modalities is a daunting task, requiring the development of a robust fusion method. The disparity of data types and structures from various modalities poses a challenge to the development of an optimal fusion technique that appropriately merges these divergent features. Secondly, non-verbal expressions of emotions can often be inconsistent or even contradictory to verbal responses, further complicating accurate emotion recognition. For example, a person may respond positively to a request while expressing sadness through their vocal tone or facial expressions. Deciphering the authentic emotional state from contradictory modalities becomes a challenge for an automatic emotion recognition system. Discerning which modality conveys the most accurate representation of the emotional state, and how to effectively filter out misleading information, adds a further layer of complexity. Finally, determining the importance of the features of the different modalities remains an open question. Determining whether all the characteristics of the different modalities should have the same importance, or whether certain characteristics should be privileged over others. Assigning an appropriate weight to these characteristics in a multimodal framework is a critical aspect that requires in-depth study and research in this field.

Numerous fusion techniques have been proposed in the field of multimodal emotion recognition including early fusion approach [18–20], and late fusion approach [21, 22]. However, current research focuses on finding an effective fusion approach that facilitates meaningful interactions between different data modalities. Previous studies have focused on designing fusion techniques or end-to-end models capable of regulating interactions between different data modalities [23–25]. Unfortunately, many of the proposed methods overlook the possibility that some data modalities may be misleading, which could adversely affect prediction accuracy. Further research is needed to address this challenge and develop a robust multimodal system capable of determining the importance of each data modality and identifying misleading information. The ultimate goal is to build a multimodal framework in which the

influence of each modality on predictions can be accurately assessed, and erroneous information can be appropriately suppressed.

1.1 Objectives

In this thesis, we aim to design a multimodal deep learning architecture for emotion recognition. This architecture takes text, image, and audio inputs and effectively learns to extract and combine pertinent information from these diverse modalities while suppressing irrelevant and misleading elements.

More specifically, our objectives are to:

- Develop a deep learning method for Emotion Recognition and Sentiment Analysis that integrates text, image, and audio inputs effectively. The proposed method should be able to extract relevant and important information from various modalities (text, image, audio)
- Design a filtering mechanism to filter out patterns that are irrelevant or may lead to misleading results, ensuring the accuracy of emotion recognition predictions.
- Test the developed method on publicly available datasets in order to validate and analyze its effectiveness.

1.2 Thesis outline

This thesis is paper-based. It incorporates our work titled "MTGR: Enhancing Emotion and Sentiment Analysis Using Gated Residual Networks," which we published in the ICPR workshop titled "Towards a Comprehensive Analysis of People: From Face and Body to Clothes" (T-CAP Workshop at ICPR 2022).

The remainder of this thesis is as follows:

- In Chapter 2, we review artificial intelligence concepts, machine learning, and the diverse techniques of deep learning. Specifically, we explore their application in tasks related to Emotion Recognition and Sentiment Analysis.
- In Chapter 3, we cover problem details, research strategy, datasets, implementation environments, proposed solutions, and key contributions.
- In Chapter 4, we introduce MTGR, a novel multimodal emotion recognition system that leverages the power of the Multimodal Transformer (MulT) and Gated Residual

Networks (GRNs). Notably, MTGR has demonstrated its effectiveness by achieving state-of-the-art performance on both the CMU-MOSI and IEMOCAP datasets.

- In Chapter 5, we present Transformers as a computational engine for tackling the Multimodal Emotion Recognition task. Moreover, we introduce diverse techniques for fusing multimodal data within a pre-trained transformer.
- In Chapter 6, a thorough exploration and comparison of the models proposed to address our Multimodal Emotion Recognition problem are presented and discussed in detail.
- In Chapter 7, we bring our thesis to a conclusion and outline potential paths for future research.

CHAPTER 2 BACKGROUND AND RELATED WORKS

In this chapter, we start by providing the essential theoretical knowledge about Machine Learning (ML) and Deep Learning (DL) for understanding their application in our work. We then review some ML and DL methods used for emotion recognition and sentiment analysis.

2.1 Machine learning background

Machine learning is the study of computer algorithms capable of enhancing performance based on learned patterns from experience, rather than rigidly following specific and predetermined instructions. Many approaches can be taken when conducting machine learning. They are traditionally divided into five broad categories depending on the data available to the learning system: 1) supervised learning, 2) unsupervised learning, 3) semi-supervised learning, 4) weakly supervised learning, and 5) reinforcement learning.

Considering our utilization of labeled datasets for emotion recognition and sentiment analysis tasks, which include multimodal input data associated with corresponding sentiment or emotion annotations, this chapter will focus solely on supervised learning, since other learning methods are out of scope.

2.1.1 Supervised Learning

The objective of supervised learning is to build a predictive model, such as a linear regression or classification model, based on prior knowledge extracted from available data. The idea is to learn an unknown function that establishes the mapping between inputs (also known as input variables or features) and outputs (also known as target variables or predicted variables). Given a set of N observations of the form $(X^{(i)}, Y^{(i)})$, $i \in 1..N$, where $X^{(i)} = (X_1^{(i)}, X_2^{(i)}, \dots, X_p^{(i)})$ is a p -dimensional vector, which is the input vector of the i -th example and $Y^{(i)} = (Y_1^{(i)}, Y_2^{(i)}, \dots, Y_k^{(i)})$, a k -dimensional vector representing the output vector of the i -th example. A learning algorithm seeks a function that maps X to Y . However, a question arises: how can we find this function? One approach involves fitting a linear model to our training data using

$$Y = g_{\omega}(X) = Y + \epsilon \quad (2.1)$$

where g_ω is a parameterized linear function, and ϵ is the error between the predicted value $g_\omega(X)$ and the real data value Y . There are many different methods to fit the linear function to the training dataset, but by far the most popular one is picking the coefficients ω^* that minimizes the most the residual sum of squares,

$$\omega^* = \operatorname{argmin}_\omega (RSS(\omega)), \quad (2.2)$$

with the residual sum squares error given by

$$RSS(\omega) = \sum_{i=1}^N \|\epsilon^{(i)}\|_2^2 = \sum_{i=1}^N \|Y^{(i)} - g_\omega(X^{(i)})\|_2^2 = \sum_{i=1}^N \sum_{k=1}^K (Y_k^{(i)} - g_k(\omega, X^{(i)}))^2, \quad (2.3)$$

where $g_k(\omega, X^{(i)})$ is the k -th component of the response $g(\omega, X^{(i)}) = (g_1(\omega, X^{(i)}), \dots, g_K(\omega, X^{(i)}))$ to the i -th data point observation $X^{(i)}$ of the input variables. And $Y_k^{(i)}$ is the k -th component of the target value for the i -th data observation.

2.2 Deep learning models

Deep Learning models are a subset of machine learning models that utilize large neural networks composed of multiple layers to learn and extract patterns and representations from data. Unlike traditional machine learning models, which often rely on predefined function models, deep learning leverages the power of neural networks. A neural network consists of interconnected neurons that cooperate to process the data and output a result. In deep learning, these networks can include different types of layers, such as fully connected layers, Convolutional Neural Network (CNN) layers, Recurrent Neural Network (RNN), etc. Generally, each type of layer was designed to handle and process different input data types. For example, CNNs are usually used to process and extract features from image data, RNNs are used for sequential data, and fully connected layers for general feature extraction or data processing.

Due to their capacity to handle large amounts of data and their ability to extract complex features, deep learning networks have achieved significant success in many domains such as Computer Vision (CV) and Natural Language Processing (NLP).

2.2.1 Perceptron

The Perceptron [26] is a single-layer feedforward neural network, considered a fundamental unit in machine learning and deep learning. It is designed to mimic the functioning of

biological neurons. It consists of interconnected processing units, often called nodes, linked together by weighted connections. Figure 2.1 provides insight into the computations occurring within the perceptron.

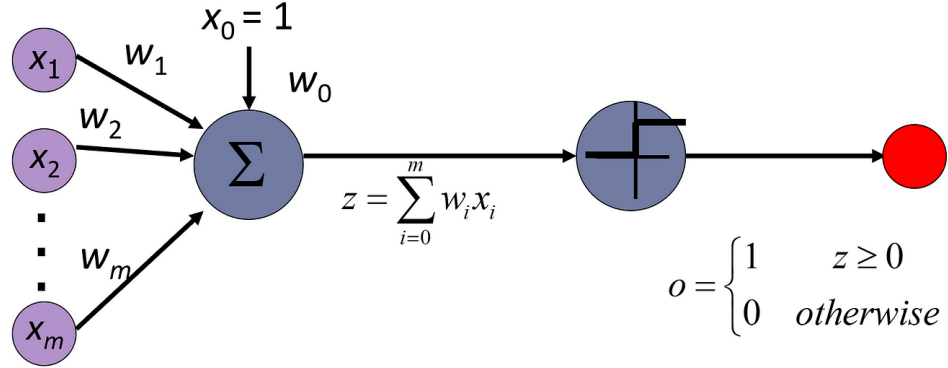


Figure 2.1 Within a Perceptron, each input is multiplied by an associated weight. The resulting weighted inputs are aggregated and processed by an activation function, determining the output. The goal is to seek the ideal weight values that minimize the error [1] © Roi Yehoshua, 2023.

A perceptron works in the following way:

- The inputs are multiplied by a matrix of weights
- Each neuron has a unique number associated with it called bias. This bias is added to the weighted sum of input.
- The result is going into an activation function. The output of this activation function will determine if the neuron gets activated.
- Every neuron that is activated will pass the information to the next layer, and this will continue until the last layer.
- The weights and bias are continuously adjusted to reduce the error function during training.

The Perceptron is primarily suitable for addressing linear problems. To tackle more complex, non-linear challenges, one can employ a multi-layer perceptron (MLP), also known as a feedforward neural network. This network structure comprises an input layer, an output layer, and one or more hidden layers.

2.2.2 Convolutional Neural Network

Convolutional Neural Networks [27], often abbreviated as CNNs or ConvNets (refer to Figure 2.2), represent a specialized category of deep neural networks primarily utilized for image-related tasks. The term "convolution" refers to the filtering operation inherent to these networks, enabling the extraction of spatial features from images. Interestingly, CNNs have demonstrated their adaptability to capture local features within sequential data, including audio and text. Thus, they have achieved notable success not only in image processing but also in the domains of natural language processing and audio processing.

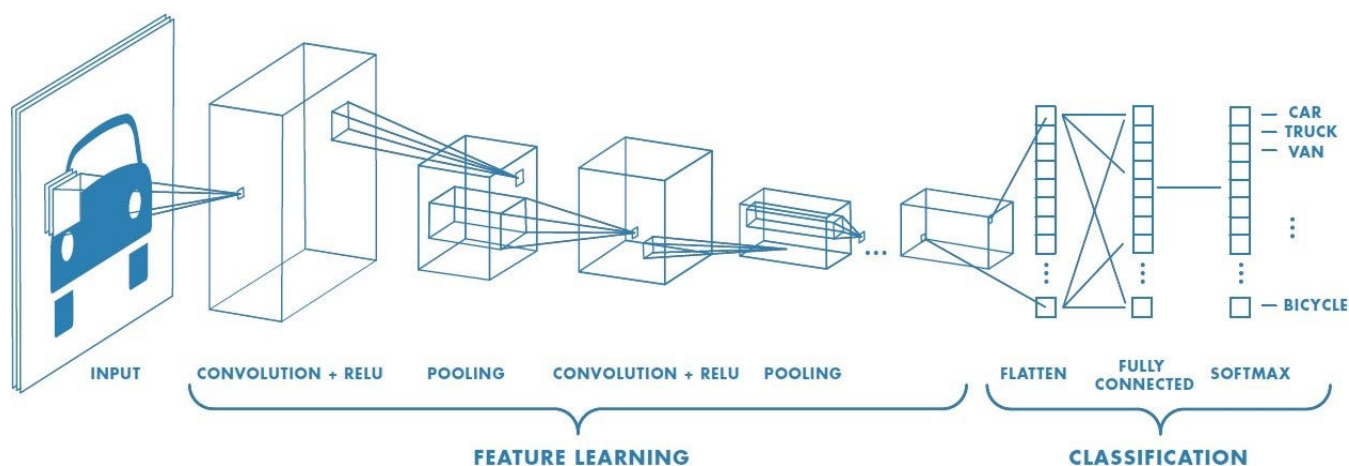


Figure 2.2 A standard convolutional neural network (CNN) architecture comprises feature learning layers, typically consisting of a combination of pooling and convolution operations applied to the input image, as well as a classification layer, typically implemented as a fully connected layer [2] © Sumit Saha, 2018.

A CNN contains:

- Convolution layer: is the most important layer where the convolution operation happens. The convolution operation (figure 2.3) is the dot product between the filter weights and the input values. Using these filters, we aim to simplify the input so it can be better processed and understood. The objective of the convolution operation is to extract high-level features such as edges, from the input data.

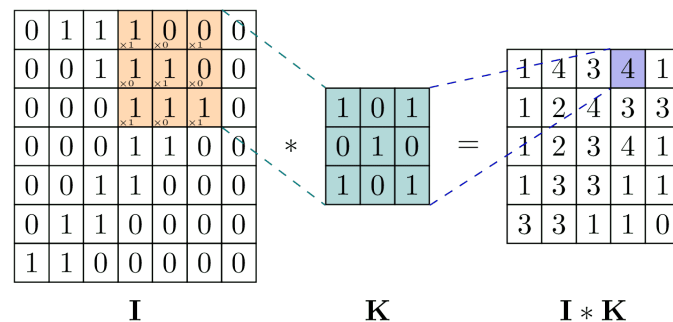


Figure 2.3 In a convolution operation, a kernel, also referred to as a filter, slides over the entire input image region by region. Within each region, the values of the input image and the kernel are subjected to element-wise multiplication. The outcomes of these multiplications are aggregated by summation [3] © Copyright 2021, Janosh Riebesell.

- **Pooling Layer:** it downsamples or reduces the size of a particular feature map. This makes the processing faster as it will reduce the number of parameters the network needs to process. This also can be seen as extracting the dominant feature. There exist many types of pooling (figure 2.4), such as the max-pooling and the average pooling. Max Pooling returns the maximum value from the portion of the image covered by the kernel. On the other hand, average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

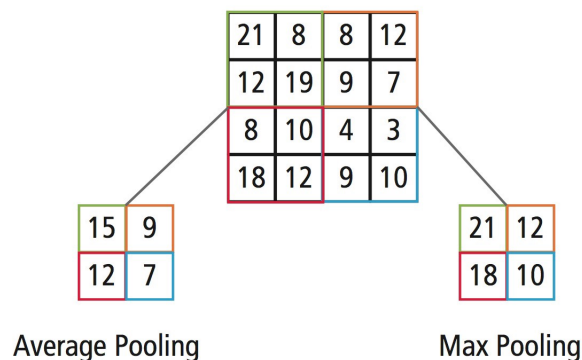


Figure 2.4 During the pooling operation, a window slides over the input image region by region. In the case of max pooling, the highest value within the window is selected at each region, while average pooling involves computing the mean value of the window elements [4] © 2018, David S. Batista.

2.2.3 Recurrent Neural Networks (RNNs)

The previous Neural Networks were primarily designed for handling static or non-sequential data, which do not exhibit specific temporal dependencies. In contrast, Recurrent Neural Networks (RNNs) [28] have emerged as a potent solution to tackle sequences of variable length organized in a particular chronological order. They have an internal memory mechanism, enabling them to utilize information from previous inputs when processing the current data. RNNs achieve this by establishing a recurrent relation over the elements within these sequences, enabling to capture temporal dependencies. RNNs have found diverse applications, particularly in tasks such as language modelling and speech recognition where we need to process sequential data.

RNN is defined with the help of the following:

- A times step of input vectors: $(X_1, \dots, X_{t-1}, X_t, \dots)$, meaning that at each time step t , we have as inputs X_t .
- A times step of states vectors $(S_1, \dots, S_{t-1}, S_t, \dots)$, meaning that at each time step t we have as hidden layers the vector S_t . S_t defines the state of the neural network at time t .
- A times step of output vectors $(O_1, \dots, O_{t-1}, O_t, \dots)$, it means that at the time step t we have as output O_t .

RNNs get their name because of the following recurrent relationship:

$$S_t = f(S_{t-1}, X_t). \quad (2.4)$$

It means that the actual state depends on the actual inputs X_t and the previous state S_{t-1} . The function f summarizes a parametric linear transformation of X_t and S_{t-1} with parameters (also called weights) W and U , and an activation function g so the relationship will be as follows:

$$S_t = g(S_{t-1} * W, X_t * U). \quad (2.5)$$

The output O_t is a transformation of the state S_t with weight vector V and an activation function $\tilde{g}$:

$$O_t = \tilde{g}(S_t * V). \quad (2.6)$$

For most of the cases, the activation functions are nonlinear functions, for example, Sigmoid and tanh. In the case where the outputs are the probabilities of classes (in a classification

problem), we often use the Sigmoid function (for a two-class problem) or the Softmax function (for a multi-class problem) because their values range between 0 and 1. The part that encodes the sequence of inputs into states S_t is called the encoder and the part that decodes the states S_t into a sequence of outputs O_t is called the decoder.

Figure 2.5 below shows the basic architecture of an RNN. Figure 2.6 shows different combinations of sequences that RNNs could perform.

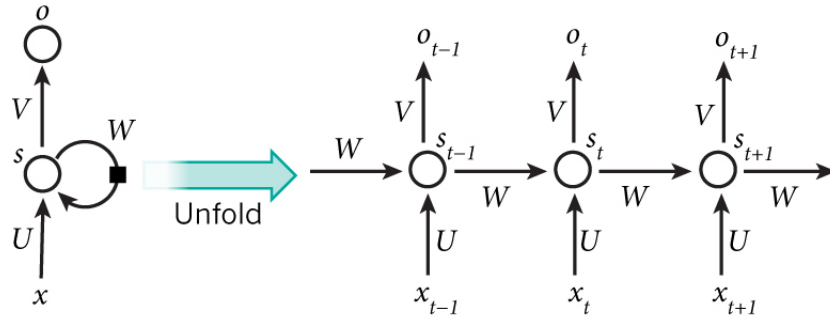


Figure 2.5 The RNN's output at each time step t is determined by the current input X_t and the hidden state from the preceding time step S_{t-1} . [5] © Aman Singh, 2021.

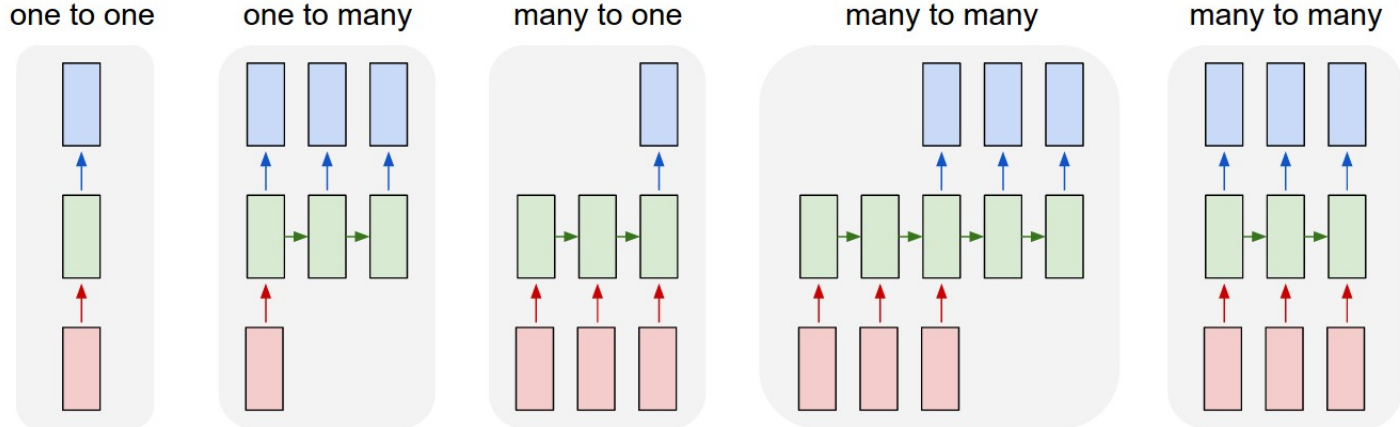


Figure 2.6 From left to right: "One to One" RNN architecture involves a single input connected to a single output. "One to Many" occurs when a single input generates multiple outputs. "Many to One" denotes a scenario where the RNN takes a sequence of inputs and predicts a single output. "Many to Many" when the RNN predicting a sequence of outputs based on a sequence of inputs. "Many to Many (2)" refers to a specific case in which each element in the input sequence of the RNN corresponds directly to a specific element in the output sequence [6] © Isheunesu Tembo, 2019.

2.2.4 Long-Short-Term-Memory (LSTM)

An LSTM Neural Network [29] is a particular instance of an RNN, designed for making decisions about whether to retain or discard past states and regulating the extent of earlier information considered during the current state. Unlike the standard RNN, the LSTM model is characterized by its increased complexity. Here, each unit at a given time t is treated as a cell, and its state at time t is denoted as C_t . Figure 2.7 describes a cell of an LSTM Neural Network. To gain a comprehensive understanding of this model, let us establish the following definitions:

- A cell memory transferred at time step t , h_t : It summarizes the actual cell state C_t and the actual output O_t by the formula:

$$h_t = O_t * \tanh(C_t) \quad (2.7)$$

- The input to be added at time step t , $\hat{C}_t$: It summarizes the actual input X_t and the previous output from the memory h_{t-1} with $\tanh$ as activation function:

$$\hat{C}_t = \tanh(W_c * h_{t-1} + U_c * X_t + b_c). \quad (2.8)$$

- The forget gate, denoted as f_t : Employs a sigmoid activation function σ that operates within a range of 0 to 1. Its primary function is to determine, based on the information received from the prior memory state h_{t-1} and the current input X_t , whether the current state cell C_t should retain or discard the influence of the preceding state cell C_{t-1} (if it is close to 1 it should be retained if it is close to 0 it should be forgotten):

$$f_t = \sigma(W_f * h_{t-1} + U_f * X_t + b_f). \quad (2.9)$$

- The input gate, represented as i_t : Employs a sigmoid activation function denoted by σ , operates within the range of 0 to 1. Its primary function is to determine, based on the information received from both the previous memory state h_{t-1} and the current input X_t , whether the current state C_t should either dismiss or include the current input. Values closer to 0 signify a preference for ignoring, while values closer to 1 indicate a strong dependency on the current input for the updated cell state:

$$i_t = \sigma(W_i * h_{t-1} + U_i * X_t + b_i). \quad (2.10)$$

- The cell state C_t : based on the input gate and the forget gate, the actual cell state will

be:

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t. \quad (2.11)$$

- The output O_t : With an activation function g , it depends on the actual input X_t and h_{t-1} :

$$O_t = g(W_o * h_{t-1} + U_o * X_t + b_o). \quad (2.12)$$

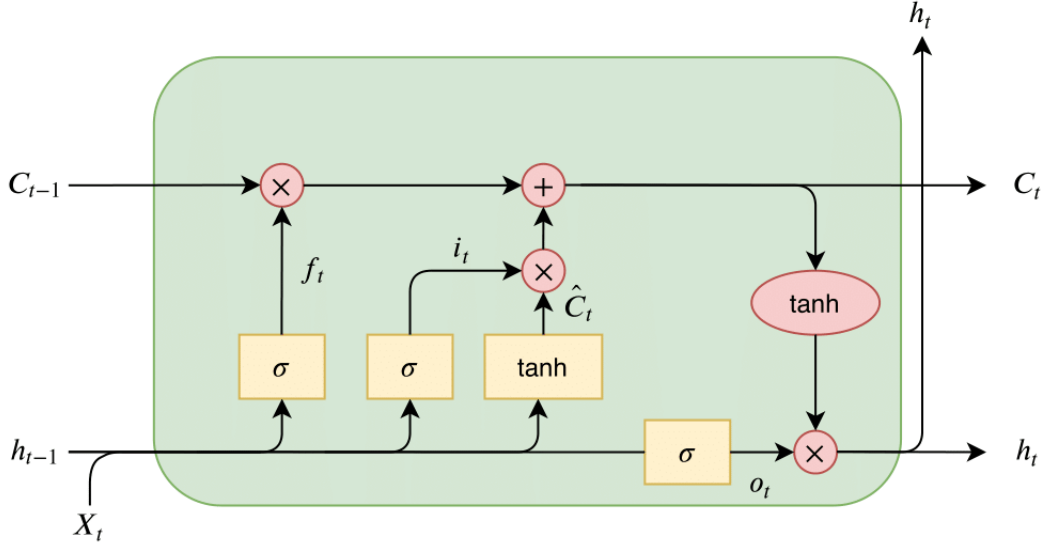


Figure 2.7 At each time step t , the LSTM cell takes the current input X_t the preceding cell memory h_{t-1} , and the previous cell state C_{t-1} as inputs. The input gate f_t applies a sigmoid function to a combination of the current input X_t and the previous memory state h_{t-1} . Similarly, the input gate i_t also employs a sigmoid function on a combination of the current input and the previous memory state. The new input value $\hat{C}_t$ is obtained as the $\tanh$ of a linear combination of the current input X_t and the previous memory state h_{t-1} . The cell state C_t is then computed based on the input gate i_t , the forget gate f_t , and the new input value $\hat{C}_t$. Finally, the output O_t is calculated based on the current input X_t and the previous memory state h_{t-1} . [7] © Thorir Mar Ingolfsson, 2021.

RNNs and LSTMs had limited success due to some major issues. First, their difficulties to process long sequences of data, which means that the model might have a problem remembering the processed information at the beginning of a long input sequence. RNNs also experience vanishing/exploding gradient problems.

2.2.5 Transformers

Transformers [30] were introduced to solve previous sequence model issues. The main component of the transformer is the self-attentive layer. For each element in the input sequence,

three vectors are calculated: the query vector Q represents the importance of an element in relation to the other elements in the input sequence, the key vector K represents the importance of an element in a specific context, and the value vector V is the representation of the element in the sequence. Figure 2.8 describes a self-attention layer. The vectors Q , K , and V are fed as inputs into the attention layer. The attention is calculated as follows:

$$attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (2.13)$$

where d_k denotes the dimension of the vectors Q and K .

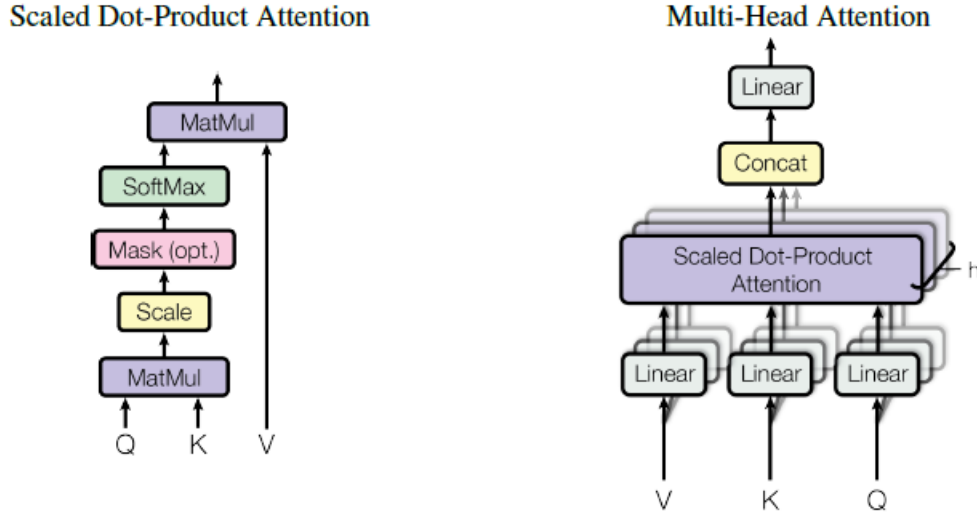


Figure 2.8 (1) On the left side, we observe the computations taking place within the scaled dot-product attention mechanism. This operation takes key (K), query (Q), and value (V) vectors as input. Initially, it computes the dot product between each query and all key vectors, resulting in attention scores. Then, these attention scores go through a scaling process, followed by the application of a softmax function to generate attention weights for each key vector. Finally, these weights are employed to calculate the weighted sum of the value vectors. (2) On the right side, we have the concept of multi-head attention, which involves the parallel calculation of attention scores using multiple attention heads. [8] © Vaclav Kosar, 2022.

2.2.6 Gated Residual Networks

Gated Residual Networks (GRNs) [31] are neural networks commonly incorporated into deep architectures to capture relationships within datasets and enhance the flow of pertinent information in these architectures. The GRN primarily consists of a residual connection and a gating mechanism, as illustrated in figure 2.9.

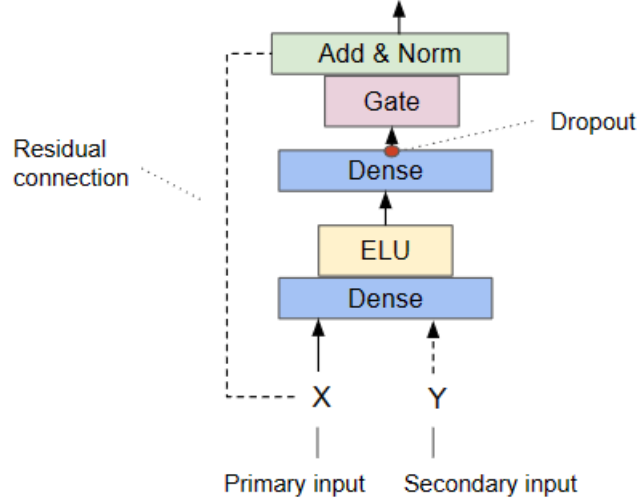


Figure 2.9 A Gated Residual Network layer takes two inputs: X as the primary input and Y as an optional secondary input. It includes skip connections and gating layers capable of dynamically learning how to combine the two inputs.

In a GRN layer, two inputs are utilized: a primary input and an optional secondary input. The layer dynamically learns the optimal combination of these inputs to modulate the information flow.

A GRN is defined as follows:

$$GRN_{\omega}(X, Y) = LayerNorm(X + GLU_{\omega}(\eta_1)) \quad (2.14)$$

$$\eta_1 = W_{1,\omega}\eta_2 + b_{1,\omega} \quad (2.15)$$

$$\eta_2 = ELU(W_{2,\omega}X + W_{3,\omega}Y + b_{2,\omega}) \quad (2.16)$$

$$GLU_{\omega}(\gamma) = \sigma(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega}). \quad (2.17)$$

Here, X is the primary input, Y is the secondary input, the Exponential Linear Unit (ELU) serves as the activation function, $LayerNorm$ is the function for layer normalization, $W_{i,w}$ and $b_{i,w}$ represent the i th weight and bias, and ω denotes the weight sharing (In the scope of

our project weight sharing is mostly based on the specific modality of data being processed). The Gated Linear Unit (*GLU*) is employed, where σ denotes the sigmoid function, and $\odot$ denotes the elementwise Hadamard product.

If the value of $(W_{2,\omega}X + W_{3,\omega}Y + b_{2,\omega})$ is significantly greater than 0, *ELU* functions as an identity function; otherwise, it generates an output. When the output of *GLU* approaches 0, the *GRN* layer will learn the identity mapping. Alternatively, if the output deviates from 0, the *GRN* layer adaptively learns to combine the two inputs. This adaptive combination aids in facilitating the learning of complex interactions and suppressing unnecessary components within a deep learning architecture.

2.3 Losses and hyperparameters

In this section, we provide a concise overview of the hyperparameters crucial for managing the training and performance of our models.

Loss: A loss function is used to evaluate the output value of the model. It measures how well the model can predict the result by comparing the output value and the ground truth expected value. If the prediction of the model is perfect, the loss will be zero; otherwise, it will be greater. The goal is to find a set of weights and biases that minimize the loss.

Learning rate: It determines how much to change the model parameters each time these parameters are updated while moving toward a minimum of a loss function. The lower the value of the learning rate, the slower we go down the slope (see figure 2.10).

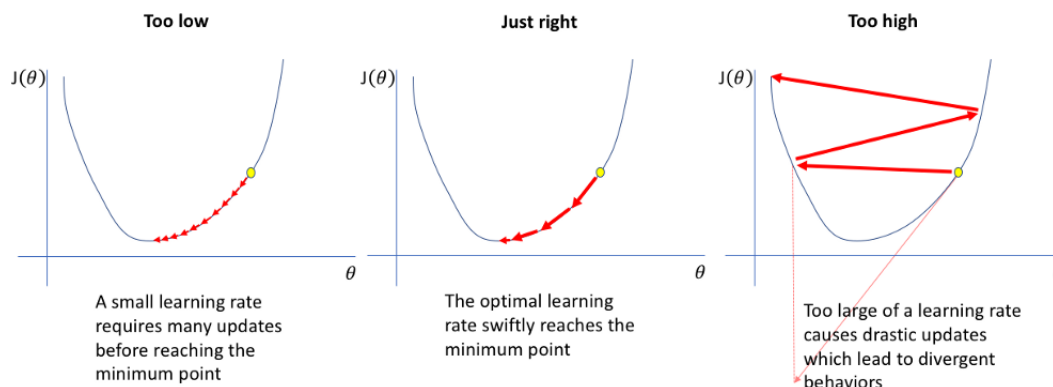


Figure 2.10 "Too low": When the learning rate is set too low, a large number of weight updates is required to minimize the error function, which results in slow convergence. "Just right": The learning rate is appropriately set. "Too high ": If the learning rate is too high, the model parameters are updated with large steps, preventing proper convergence of the model [9] © Jeremy Jordan, 2018.

Number of layers and number of nodes: To manage the structure and depth of the network, we have the flexibility to adjust both the number of layers and the number of nodes embedded within each hidden layer. The number of layers as well as the number of nodes in each layer exert a significant impact on the model performance. An insufficient number of layers or nodes may lead to underfitting, resulting in a failure to capture complex patterns, while an extensive number of layers or a large number of nodes within the hidden layers can provoke overfitting, where the model excessively customizes itself to the training data, losing its ability to generalize. In cases of deep architectures, the employment of regularization techniques becomes essential to mitigate the risk of overfitting.

Dropout: It is a regularization technique, which is commonly employed in neural networks to mitigate the risk of overfitting, especially in deep architectures. It operates as a network regularization method by randomly selecting nodes and assigning each a probability of being ignored during the training process. This random dropout of nodes helps to learn independent representations, thus improving the ability of the model to generalize.

Number of epochs: It represents how many complete passes the entire training dataset makes through the network during the training process. An excessively high number of epochs can induce the model to overfit the training data. Conversely, A very small number of epochs may trigger underfitting issues.

Batch size: It represents the amount of training samples used at each iteration to modify the model weights. When the batch size is quite small, the model can capture patterns from each example, but the training process will be considerably time-consuming. Conversely, if the batch size is excessively large, the model may lose its ability to generalize efficiently, leading to overfitting problems.

2.4 Related work

Emotion recognition and sentiment analysis have been for years challenging tasks, and they are fluidly following the advancement in Artificial Intelligence. The initial phase of this evolution involved the statistical method phase that employed simple machine learning methods. Subsequently, the field transitioned into the current state-of-the-art phase, which relies on deep learning to harness the full potential of advanced computer vision and natural language processing methods. In our review, we categorize our analysis based on the type of input data, outlining the approaches employed for textual, image, and audio data. Finally, we detail significant state-of-the-art techniques employed for multimodal data.

2.4.1 Text-based Methods

In this section, we explore text-based methods for emotion recognition and sentiment analysis. Before machine learning, lexicon-based approaches were one of the first methods used to detect emotions, such as keyword-based algorithms [32] that rely on the selection of a set of emotional keywords to classify text. The problem with these traditional methods is that the keyword may not be present in the sentence, or the emotion it conveys may vary depending on the language, context, and domain of the text.

Machine learning algorithms have been introduced to solve the above problems. For example, Yong-Soo et al. [33] have proposed to use a knowledge-based artificial neural network (KBANN). They train KBANN to detect emotions using approximate domain knowledge expressed in rules (hierarchical features), meaning that the model will be trained to learn this hierarchical representation of features and how to detect emotions from them.

Balabantaray et al.’s research [34] involved the development of an emotion classifier that utilized multi-class Support Vector Machine (SVM) kernels to train and categorize tweets. The primary objective of this classifier was to assign emotional labels to tweets based on their content. In this study, they utilized multi-class SVM kernels, making the model suitable for multi-label emotion classification. The authors represented each sentence as a vector, incorporating various features present in the sentence along with their respective feature

weights. These features encompassed three distinct categories:

- Syntactic features, which encoded the grammatical structure;
- Semantic features, responsible for capturing the meaning;
- Contextual features, designed to capture contextual information related to emotion words.

By combining these diverse features, they trained a multi-class SVM model, achieving an accuracy rate of 73.24%.

Hasan et al. [35] conducted a study in which they trained and compared the performance of four different classifiers, namely the Support Vector Machine, Naive Bayes (NB), Decision Trees, and K-Nearest Neighbors, for the task of emotion detection in tweets. Each classifier was specifically trained to analyze tweets, which were represented through a combination of diverse features. These features include:

- Unigram Features: In this approach, tweets were transformed into vectors of unigrams. To reduce the feature size, the authors designed a domain-specific dictionary using emotional lexicons;
- Emoticon Features: Emoticons, often rich in emotional content, were incorporated as features to enhance emotion classification;
- Punctuation Features: Punctuation marks were harnessed as additional features to provide context and meaning to the expressed emotions in tweets;
- Negation Features: A separate feature was introduced to account for negation, addressing potential misinterpretations caused by negated phrases.

The performances of the classifiers were evaluated based on precision, recall, and F-measure. The results indicated that when considering all features or only the unigram features in the case of the SVM classifier, Decision Trees, K-Nearest Neighbors, and Support Vector Machine achieved the highest accuracy, all surpassing the 90% mark. Notably, K-Nearest Neighbors and Support Vector Machine demonstrated a faster execution time compared to the Decision Tree classifier.

Similarly, Armin et al. [36] proposed a model based on Logistic Regression (LR) and random forest. Their idea was to train an LR model to select the features to be used by the random forest for classification.

Tremendous interest in deep learning has emerged in recent years. The application of deep learning methods in almost all sub-fields of computer vision and natural language processing has gained much attention. This is particularly due to the enormous potential of CNN to extract spatial features. As well as to the efficiency and the progress of sequence models (RNN, LSTM, Transformers, etc), showing very promising results in classifying, processing, and making predictions based on time series data. Numerous deep learning techniques have been proposed based on these models.

Tripto et al. [37] conducted the development and comparative analysis of two deep learning models for emotion classification. The initial model incorporates an LSTM layer followed by a dense layer with softmax activation. The second model utilizes a 1D convolutional layer followed by a dense layer and a softmax layer for emotion classification. Both models receive sentence embeddings as inputs. In the preprocessing step, each word within a sentence is represented as a vector using Word2Vec. Subsequently, each sentence in the dataset is converted into a one-hot encoding vector of length D , where D corresponds to the vocabulary size. To generate these embeddings, they employed a feed-forward network with a single hidden layer composed of m nodes and a linear activation function. This network takes the one-hot encoding vector of a sentence as input and computes the probability of placing the respective word within that sentence. Consequently, this network creates a weight matrix in which each row represents a word, and the length of the word vector in the sentence is now m . These newly obtained sentence embeddings serve as the input for their proposed classification models. In their study, Tripto et al. observed that the LSTM and CNN approaches outperformed traditional machine learning methods, such as SVM and NB. Notably, the LSTM model exhibited slightly superior performance compared to the CNN model.

Batbaatar et al. [38] introduced the SENN deep neural network framework, comprising a combination of a Bidirectional Long Short-Term Memory (BiLSTM) and CNN. The BiLSTM component was employed to model text semantic information by capturing hidden states for each word in both forward and backward directions, while the CNN was designed to extract emotional features. To test the model capability, they used various word embeddings, including Word2Vec, GloVe, and FastText, to capture semantic word relationships, and emotional word embeddings to extract emotional features. Thus, the word embedding layer encodes sequence representations into two embedding vectors: one for emotional word embedding and the other for semantic word embedding. Semantic word vectors were fed into the BiLSTM, while the CNN received the emotional word vectors. Finally, semantic and emotional features are concatenated and fed into a feedforward network to predict emotion. When compared to machine learning models and various other deep learning models like CNN, Gated Residual Unit (GRU), Bidirectional GRU (BiGRU), LSTM, BiLSTM, Recurrent Convolutional Neu-

ral Network (RCNN), and Conv-LSTM utilized for emotion classification, their SENN model demonstrated best performance across nine diverse datasets.

Wenhao et al. [39] have proposed a BERT-based model designed to classify emotions in tweets by combining both general and domain-specific knowledge. To achieve this, they employed a Twitter-specific preprocessor to handle tweets, automatically annotating domain-specific expressions by introducing special tokens. For instance, in the tweet "haaaappy birthday!!," a special token '<elongated>' is added after the word 'happy' to enhance the detection of positive sentiment. Their approach initially involved fine-tuning BERT to extract general features from Twitter data. Additionally, they introduced a token pattern detector, implemented as a CNN, which is trained to identify token patterns relevant to emotion classification. This CNN was trained from scratch using training data processed with the Twitter-specific preprocessor. The output from these detectors is used as the domain-specific features for each sentence. Finally, they integrated the features derived from BERT with the domain-specific features by concatenating them, and they utilized a linear scoring layer for emotion prediction. Their results proved that the integration of both types of knowledge significantly enhanced emotion classification performance.

Hassan et al. [40] proposed a deep learning framework to learn emotion-specific associations by learning label correlations. The labels and the input sentence are fed into an encoder BERT to obtain hidden representations for both words and emotion classes. Finally, the hidden representations are fed into a feed-forward network to obtain a score for each token, where the scores for the label tokens are used as predictions for the corresponding label.

2.4.2 Image-based Methods

In this section, we explore image-based methods for emotion recognition and sentiment analysis. Traditional approaches are mainly based on three different stages: image pre-processing, feature extraction, and emotion classification. At each stage, it is essential to select the appropriate machine learning algorithm or method. For instance, Cornejo et al. [41] have proposed a facial expression-based emotion recognition system that is robust to occlusion. They first perform a robust principal component analysis (RPCA) to reconstruct the occluded facial expression. They then perform fiducial point detection to extract facial expression features. They apply Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce the vector space of extracted features. Finally, they use K-NN (K-nearest neighbor algorithm) and SVM classifiers to recognize the expressions.

Gomathi et al. [42] have proposed a neuro-fuzzy-based system for automatic facial expression recognition. Their method first segments the image into three regions: the upper region

(eyes and eyebrows), the middle region (nose), and the lower region (mouth). Next, they use the Local Binary Pattern (LBP) to extract the feature distributions of these regions and represent them as 7-bin histograms. Finally, these histograms are aggregated on the basis of a weighted average and fed into the Multiple Adaptive Neuro-Fuzzy network (MANFIS) to recognize emotions.

In [43], the authors proposed to recognize emotions using LPTP (Local Principal Texture Pattern) features. They calculate LPTPs at each pixel by extracting the principal directions of the local neighborhood and then encoding the intensity differences in these directions. Finally, they use the SVM classifier to classify the emotions and evaluate their feature extraction technique.

The rise of deep learning methods in computer vision has reduced the effort required by machine learning models and overcame the problems of manual image processing and feature extraction, offering the possibility of building end-to-end models capable of image processing, feature extraction, and emotion detection. For instance, Pramerdorfer et al. [44] have utilized and compared different CNN architectures to predict emotions from a single image. They reviewed six shallow CNN architectures. They then compared them with state-of-the-art computer vision architectures, including VGG, Inception, and ResNet, and proved that using deep CNN architectures significantly improved the results.

Transformers have also been used in computer vision. The first vision transformer (ViT) [45] was inspired by the vanilla transformer used in NLP, with a few modifications so that it could be applied to image classification tasks. Various vision transformer architectures have been proposed for image-based emotion recognition. For instance, Wen et al. [46] have proposed a network called: Distract Your Attention Network (DAN). Their method consists of three elements: a feature clustering network using a residual network as a backbone, a multi-headed attention network (MAN) that uses both spatial and channel-based attention to effectively model feature interactions, and finally a proposed attention fusion network (AFN) to distract these attentions to multiple locations before fusing the feature maps.

Xue et al. [47] proposed a framework that combines both a CNN and a transformer; in addition, they introduced two attention pooling modules, namely Attentive Patch Pooling (APP) and Attentive Token Pooling (ATP). They first used a CNN to extract feature maps from images, then the extracted features were used to calculate an attention map to be used by the APP module to select important patches based on their attention values. The selected patches are then fed into the ViT where its blocks have been modified by adding ATP which helps to reduce irrelevant tokens.

2.4.3 Audio-based Methods

Acoustic features are also used to detect emotions. Wu et al. [48] used acoustic-prosodic features and semantic labels for emotion recognition. These acoustic-prosodic features encompass various speech characteristics, such as pitch, intensity, speech rate, tone, and intonation, among others. Their methodology relied on utilizing Gaussian Mixture Models (GMM), SVM, and MLPs as basic classifiers. To fuse these classifiers and achieve confidence in emotion recognition for the acoustic features, they incorporated a meta-decision tree (MDT). Furthermore, they applied a Maximum Entropy model (MaxEnt) to establish the connection between emotional states and emotion association rules (EARs) extracted from semantic labels. To integrate the results of both acoustic and semantic recognition, they implemented weighted product fusion.

In the work presented by Petrushin et al. [49], the authors created an application designed for call centers. In their investigation, they focused on acoustical variables, which encompassed fundamental frequency, energy, speaking rate, the first three formants, and their respective bandwidths. For each of these variables, they calculated several descriptive statistics, including the mean, standard deviation, minimum, maximum, and range. In total, there were 43 features computed for each spoken utterance. They employed a feature selection algorithm to select the top 14 features that proved most relevant for predicting emotions. Within their application, the authors explored different classification methods, which included the use of K-NN, Artificial Neural Networks (ANN), and an ensemble of ANNs. Their aim was to study the effectiveness of these classifiers to recognize emotions conveyed through speech. Their results show that the ensemble of ANNs performed better compared to the other classifiers.

Keren et al. [50] introduced a Convolutional Recurrent Neural Network (CRNN) for the classification of emotions based on audio recordings. In their approach, they create a sequence of windows from the input data, with each window comprising a series of consecutive frames. These windows are then processed frame by frame through an RNN. The feature vector representing a window can be derived as the mean or maximum values from the RNN hidden states or as the vector of the hidden state from the final frame in the window. To mimic the behavior of a convolutional layer, they utilize the same RNN layer with consistent parameters to compute local features for all the windows. Their model was trained using log mel-filter banks, extracted from each utterance, as the input data representation. They employed various RNN variations, including LSTM and BiLSTM. Their results demonstrated that these RNN-based models outperformed traditional CNN in the context of emotion classification.

More recently, attention-based models have been used in speech emotion recognition due to their ability to robustly model acoustic features in long input sequences by focusing on the

parts that contain emotional information. For instance, Mirsamadi et al. [51] introduced a speech emotion recognition network that integrates an RNN with a weighted pooling approach that employs local attention. Their model was trained to recognize emotions at the utterance level, where each utterance consists of multiple frames. They derived audio feature inputs from each frame. In their experiments, they compared the use of raw spectrogram representations with hand-crafted low-level descriptors (LLDs), commonly employed in sentiment emotion recognition. To enhance the RNN focus on the most emotionally significant parts of the utterance, they introduced an attention module that computed weights for the RNN outputs. Their results revealed that the utilization of their proposed model with raw spectrogram input data yielded better results when compared to other RNN architectures (including frame-wise training, many-to-one training, and mean-pooling over time) and traditional SVMs.

In [52], the authors proposed a self-attentional CNN-BiLSTM network that consists of 2 CNN layers, 2 BiLSTM layers, and a self-attention network. The CNN layers are used to encode local features and reduce the length of the input sequence. BiLSTM layers are used to modulate global contexts. Finally, the attention layer is used to aggregate information from the BiLSTM hidden states and produce the final feature vector that encodes the input speech.

2.4.4 Multimodal-based Methods

Multimodal emotion recognition and sentiment analysis have recently become very active areas of research and have taken human-computer interaction to a new level, enabling an understanding of the emotions and expressions of the user beyond the verbal domain. Numerous multimodal emotion recognition systems have been proposed, based on AI models and techniques.

Morency et al. [53] have been the first to investigate tri-modal sentiment analysis. Their work involved proving that models that integrate visual, audio, and textual features can be used effectively to identify sentiments in videos. Their features were extracted at the utterance level to represent local changes. For text, they automatically identify the sentiment present in the text using a lexicon and then associate it with a polarity. The score obtained for the textual features is the sum of the word polarities. For visual features, they calculate the duration of the smile and the duration of the gaze away for each utterance, then normalize these two features by the total number of frames. For audio features, they used OpenEAR to automatically calculate voice pitch and intensity. Finally, they used HMMs as a classifier to classify the sentiments.

One of the challenges of multimodal classification tasks that ML models cannot handle is the ability to model inter- and intra-modality dynamics. Some deep learning systems have addressed this problem. For instance, Zadeh et al. [54] have proposed a tensor fusion network (TFN), which consists of three modules: three modality embedding subnetworks to extract a richer embedding for unimodal features, a tensor fusion layer that captures unimodal, bimodal and trimodal dynamics, and finally an inference subnetwork in which they use a fully connected neural network to perform sentiment classification.

Poria et al. [55] have proposed a hierarchical framework for context-dependent sentiment analysis. Initially, their architecture uses context-independent unimodal feature extractors to extract features at the utterance level of each modality. They then introduced context-dependent LSTMs that take context-independent unimodal features as input and extract context-dependent features. At level two of the architecture, the context-dependent features of the different modalities are concatenated and fed into an LSTM network to extract contextual multimodal features used for classification. Due to their ability to model the internal and external interactions between the different elements of an input sequence, attention-based models have also been used to integrate different data modalities.

Kumar et al. [24] have introduced a control mechanism for attention-based multimodal sentiment analysis. Their deep learning system is capable of learning interactions between different modalities controlled by learning gates. First, they extract the contextual representation of the utterance using BiGRU. Then, a self-attention layer is used to capture long-term dependencies between utterances. The main cores of their system are the cross-attention, which computes cross-attention representations for each modality, and the learning mechanism, which favors either cross-interaction or unimodal contextual representations. The BiGRU layer is then used to merge information from self- and cross-interactions. Finally, the feature vectors resulting from the BiGRU layer are concatenated and fed into a feedforward network for classification.

Tsai et al. [23] proposed a multimodal transformer for unaligned multimodal linguistic sequences. Their architecture is composed of multimodal transformers that rely on the co-attention mechanism to reinforce a target modality with the low-level features of another source modality. A self-attention-based transformer is then used, which takes as input the concatenation of outputs from cross-modal transformers that share the same target modality. Finally, the resulting features are concatenated and passed to a feedforward network for prediction.

Finally, Yang et al. [25] have proposed a cross-modal BERT that allows a pre-trained transformer like BERT to be refined on different data modalities. Their model is based on interac-

tions between two data modalities (text and audio). In the first branch of their architecture, the text modality is introduced into a pre-trained BERT. In the second branch, word-level aligned audio features are extracted. They have introduced masked multimodal attention that will dynamically adjust word weights according to text-audio interactions and finetune BERT.

2.5 Discussion

In this chapter, we have explored a range of machine learning methods, including some of the state-of-the-art deep learning methods proposed in the field of emotion recognition. Our review of the literature was organized based on the nature of input data. Specifically, we delved into text-based, image-based, audio-based methods, and multimodal methods.

While unimodal methods have demonstrated notable success, the utilization of multiple data modalities and the selection of effective fusion techniques for combining diverse features tend to yield superior results compared to straightforward unimodal methods. For example, the incorporation of multimodal transformer architectures, endowed with their co-attention module, has achieved remarkable state-of-the-art results in the realm of emotion recognition.

In our work, we drew inspiration from the capabilities of multimodal transformers in managing heterogeneous, non-aligned multimodal data. Additionally, we harnessed its co-attention mechanism, which enables the integration of two distinct modalities of data by enhancing one modality with valuable information from the other. This approach results in a richer feature representation, ideal for use in emotion classification tasks. Furthermore, we have extended the capabilities of multimodal transformers and explored pre-trained transformer models and their ability to be fine-tuned on multimodal emotion recognition tasks.

CHAPTER 3 OVERVIEW OF THE METHODS

In this chapter, we provide a concise overview of our proposed solutions and outline the notable contributions we have made. As well, we present the datasets and software and hardware environments used for the implementation of our end-to-end deep learning networks.

3.1 Research Methodology

The complexity of human emotions is a major obstacle in the field of multimodal emotion recognition. The expression of a person’s emotional state involves a mixture of verbal and non-verbal cues. Building a reliable automatic emotion recognition system requires the integration of various cues at different levels, depending on their importance in reflecting the person’s emotional state. This automated system processes various data modalities such as text, audio, and video, and strives to anticipate emotions by analyzing the information provided by these modalities. However, the use of various data modalities introduces many challenges, including the management of disparate extracted features, the risk of inconsistency between modalities, and the need to merge them efficiently in order to highlight data relevant for prediction while removing redundant information.

In the context of this thesis, video, audio, and text features are used to develop an end-to-end model for emotion recognition and sentiment analysis. The aim is to exploit and skillfully combine meaningful information from these three data modalities to accurately predict an individual’s emotional state.

In our work, we introduce and train various neural network algorithms capable of integrating features derived from various data modalities to accurately identify emotions or sentiments. Features extracted from video, audio, and text sources are fed into an end-to-end deep learning model, enabling the system to learn to extract and fuse relevant information to predict the precise emotion conveyed by these modalities. We firstly proposed to enhance multimodal transformers through the integration of Gated Residual Networks (GRNs). Specifically, GRNs are strategically employed after each co-attention operation in order to filter out unnecessary or less relevant information from other modalities of data. Using their gating mechanism, GRNs will ensure that our multimodal transformers focus more on important patterns so that misleading data does not significantly impact predictions. The use of GRNs represents a critical improvement in optimizing the performance and accuracy of multimodal

transformers in handling various types of data.

We trained MTGR on multimodal emotion recognition and sentiment analysis datasets from scratch. Despite our model remarkable achievement of state-of-the-art results across various metrics on three distinct datasets, we acknowledge that further improvements are constrained by the limited amount of public data accessible for emotion recognition and sentiment analysis. Consequently, we decided to explore the concept of pre-trained transformers, with the intention of harnessing the encoded general knowledge within their attention layers, which had been acquired during the pre-training phase. This approach allows us to absorb new patterns and representations by fine-tuning these pre-trained models on the downstream task of multimodal emotion recognition.

Thus, in our second solution, we utilized pre-trained transformers designed for language modeling tasks to investigate their adaptability in multimodal emotion recognition and sentiment analysis downstream tasks. Inspired by our prior work, MTGR, and other multimodal deep learning architectures, we proposed some fusion techniques to incorporate multimodal data into these pre-trained transformer models. Our proposed architectures are end-to-end deep learning models that employ a fusion strategy to combine various modalities and pre-trained transformers that are fine-tuned to process the different data modalities either before or after the fusion, depending on the specific fusion strategy utilized.

3.2 Dataset

Existing datasets for emotion recognition use two different annotation methods. The categorical method, which uses the six core emotions proposed by Ekman [56] (anger, disgust, fear, happiness, sadness, surprise) or the eight core emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, confidence) proposed by Plutchik [57]. The VAD dimensional method describes emotions using three dimensions: (V) Valence which measures whether the emotion is positive, negative, or neutral. (A) Arousal or Activation which measures a person’s level of stimulation ranges from calm to excitement. (D) Dominance, which measures a person’s level of control ranges from submission to feeling in control.

Unlike emotion recognition datasets, sentiment analysis datasets are annotated to determine sentiment polarity (positive, negative, or neutral), or to determine sentiment intensity which is also called the strength of the sentiment (from low to high).

Our research involved the identification and selection of appropriate public datasets to train and compare our models against state-of-the-art models. As a result, we chose three of the most widely recognized and extensive datasets employed in the domains of emotion

recognition and sentiment analysis: CMU-MOSEI, CMU-MOSI, and IEMOCAP datasets.

CMU-MOSEI stands out as the largest dataset for sentiment analysis and emotion recognition, encompassing over 23,500 utterance videos drawn from 1000 diverse YouTube speakers. Each utterance has three modalities of data, including language, visual, and acoustic features.

CMU-MOSI is acknowledged as one of the prominent public multimodal sentiment analysis datasets, comprising 2199 annotated videos for subjectivity and sentiment intensity.

IEMOCAP is a multimodal dataset encompassing approximately 12 hours of audiovisual data. It features categorical and dimensional annotations, including valence, activation, dominance, and more.

3.3 Software and Hardware Environments

To implement our end-to-end deep learning algorithms, we utilized PyCharm as the integrated development environment (IDE). Our programming language of choice was Python, specifically version 3.7.9. Additionally, we employed PyTorch, an open-source machine learning framework, in its version 1.11.0, to develop and train the various models. To enhance the computational efficiency, we used CUDA, a parallel computing API, in its version 11.3, compatible with our GPU NVIDIA GeForce RTX, facilitating the parallel processing of tensors.

3.4 Contributions

The primary contributions of this thesis encompass the following:

- A multimodal emotion recognition system based on Multimodal Transformer (MulT) and Gated Residual Networks (GRNs) is developed which achieved state-of-the-art results on CMU-MOSI and IEMOCAP datasets.
- Introducing and developing pre-trained multimodal transformer architectures by leveraging transfer learning and the capabilities of pre-trained transformers. These architectures employ various approaches to combine data modalities.

CHAPTER 4 ARTICLE 1: MTGR: IMPROVING EMOTION AND SENTIMENT ANALYSIS WITH GATED RESIDUAL NETWORKS

Conference: *ICPR 2022 International Workshops and Challenges*

Authors:

Rihab Hajlaoui: LITIV Laboratory, Polytechnique Montréal, Montréal, Canada

Guillaume-Alexandre Bilodeau: LITIV Laboratory, Polytechnique Montréal, Montréal, Canada

Jan Rockemann: Airudi, Montréal, Canada

Submission date: *20 May 2022*

Publication date: *30 July 2023*

Contribution:

I contributed to this work on literature review, experimentation, examination, design, and implementation of the method, as well as the writing phase to present MTGR, a transformer-based model for emotion recognition and sentiment analysis.

Abstract

In this paper, we address the problem of emotion recognition and sentiment analysis. Implementing an end-to-end deep learning model for emotion recognition or sentiment analysis that uses different modalities of data has become an emerging research area. Numerous research studies have shown that multimodal transformers can efficiently combine and integrate different heterogeneous modalities of data, and improve the accuracy of emotion/sentiment prediction. Therefore, in this paper, we propose a new multimodal transformer for sentiment analysis and emotion recognition. Compared to previous work, we propose to integrate a gated residual network (GRN) into the multimodal transformer to better capitalize on the various signal modalities. Our method shows an improvement of the F1 score and the accuracy results on the CMU-MOSI and IEMOCAP datasets compared to the state-of-the-art results.

Keywords — Multimodal fusion, Image and video processing, Natural language processing

4.1 Introduction

Emotions are expressed differently from one person to another. Most people communicate emotions both verbally (natural language), and non-verbally (facial gestures, tone of voice, etc.) to different degrees depending on circumstances and personalities. Acoustic and visual cues thus can provide additional knowledge to help understand human behaviours and significantly improve the prediction performance of an emotion recognition system. However, combining and integrating different heterogeneous modalities of data into an end-to-end deep learning model is not trivial due to two main reasons:

1. The non-aligned nature of features across modalities. For example, a face gesture at the time step i may relate to a word spoken at another time step j (where $i \neq j$).
2. The heterogeneity of the data across the different modalities. The features of the different modalities are extracted using different tools and with various representations.

To address the above issues, prior studies proposed different strategies to fuse different modalities of data. For instance, some of them rely on the early fusion approach to combine the feature vectors of the different modalities [18–20]. Other works used late fusion strategy to combine the results of unimodal classifiers [21, 22]. More recently, Multimodal Transformers have been proposed by many researchers and they have shown their effectiveness in dealing with the above issues. Multimodal transformers use a module called crossmodal attention (co-attention). This module will learn the attention across two modalities: the target modality, which is the main input to the model that is used to predict the sentiment/emotion, and the source modality, which is the second input to the model that is used to enhance the features of the target modality. Unlike the self-attention mechanism, the co-attention will compute a new representation for the target modality. This new representation contains information coming from the source modality.

However, we noticed that sometimes the reinforced features contain information from the source modality that might affect negatively the prediction. For example, non-verbal communication sometimes conveys meaning by contradicting verbal communication instead of reinforcing or substituting for it. *Following this observation, we propose to add a GRN (Gated Residual Network) that allows the network to have the flexibility to use or suppress information received from other modalities.*

Therefore, we propose a multimodal transformer with a gated residual network (MTGR) for emotion recognition and sentiment analysis. We use an end-to-end multimodal transformer (MulT) similar to Tsai [23] that uses co-attention, and improved their model by adding a GRN in the multimodal transformer modules. Our proposed architecture allows the multimodal transformer to minimize the contribution of unnecessary information sent from other modalities. The GRN will act as a filtering mechanism that facilitates the flow of information between the layers of the multimodal transformer. Results show that we obtain a significant improvement over the original MulT model for the CMU-MOSI and IEMOCAP datasets.

4.2 Related Work

The sentiment analysis and emotion recognition fields have observed and confirmed the effectiveness of combining different modalities of data. Many solutions have been proposed to fuse multimodal information and they can be categorized into two types of fusion:

1. **Late fusion (decision-level fusion):** consists in training several classifier networks independently, each classifier being assigned to a single modality. The final decision is obtained by applying, for instance, a weighted sum fusion or a weighted product fusion on the different results [21, 22]. Models based on this strategy are not able to learn inter-modality and intra-modality interactions;
2. **Early fusion (feature-level fusion):** In this strategy, the features of different modalities are merged into a global feature vector. Many early fusion methods can be applied, for instance, a simple concatenation [18–20], an element-wise sum, or an element-wise product [58]. Despite the simplicity of these methods, they are unfortunately not able to explore in-depth both inter-modality and intra-modality interactions.

Recently, more advanced methods were proposed. For example, Li et al. [59] proposed an outer product through the time where all intra- and inter-modality dynamics can be captured. Bagher Zadeh et al. [60] suggested Graph-MFN, which is based on a dynamic fusion graph that can learn crossmodal interactions and can effectively choose the appropriate fusion graph based on the importance of each n -modal dynamic during inference. Wang et al. [61] proposed a Recurrent Attended Variation Embedding Network (RAVEN) that is able to capture the dynamic between words and non-verbal behaviours by shifting the word representations based on non-verbal cues.

However, most of these previous works do not address the problem of feature misalignment between modalities. Other recent methods have explored the use of transformers to learn multimodal representations [23, 62] and they have been shown to be effective in dealing with unaligned multimodal language sequences. In Tsai et al. [23] and Delbrouck et al. [62], both of the proposed transformers rely on crossmodal attention (co-attention) that will learn the attention across two modalities. In other words, this module will reinforce the features of a target modality x with those from another source modality y . To do so the key K and the context C of the self-attention are calculated from the features of the modality y instead of those of the modality x . Instead of using only the co-attention, Delbrouck et al. [62] propose the idea of a modular co-attention which consists of calculating the attended features of a primary modality using the self-attention and enhancing the features of a secondary modality by those from the primary modality using the co-attention. The co-attention operation in these proposed methods leads to a new feature representation of the target modality that always contains information coming from other modalities. However, this information might be unnecessary or might contradict the meaning conveyed by the features of the target modality.

4.3 Methodology

Based on the previous works, we determined that sometimes the reinforced features might contain information from the source modality that is detrimental to the prediction. This motivated us to propose a method with more flexibility to use or suppress information received from other modalities. To do so, we propose to integrate a Gated Residual Network (GRN) [31] after the co-attention operation of a crossmodal transformer, such as the one used in [23].

Fig. 4.1 gives an overview of our proposed method.

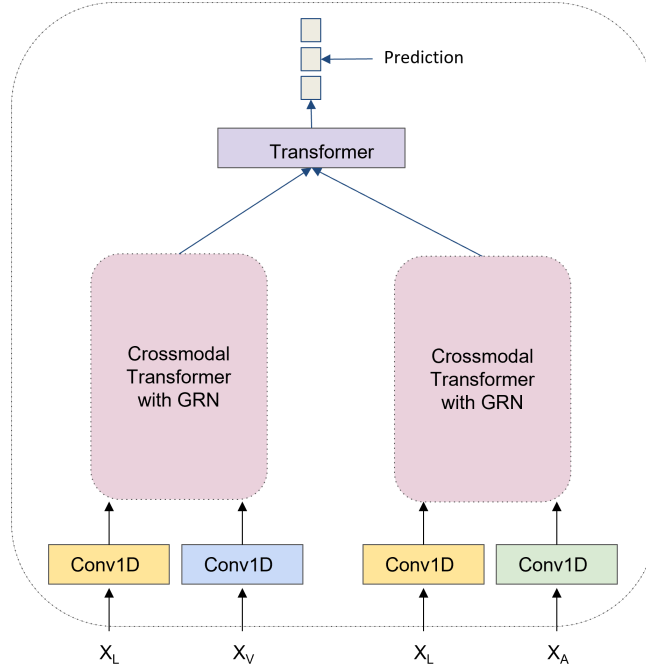


Figure 4.1 Overview of the method. The main components of the multimodal transformer are the crossmodal transformers which capture the dynamics between two modalities of data. The outputs of the crossmodal transformers are concatenated and passed through a self-attention transformer to capture temporal information for the prediction.

4.3.1 Problem Definition

We consider three different modalities of data L (Linguistic), A (Acoustic), and V (Visual). Our goal is to recognize sentiments or emotions based on these three modalities. Given the extracted features of each modality that are denoted as X_L , X_a , and X_v respectively, these feature sequences are going to be the inputs for the multimodal transformer that mainly consists of two crossmodal transformers. Each crossmodal transformer aims to modulate the dynamics between two different modalities of data to improve the prediction for a sentiment analysis or an emotion recognition task. Crossmodal transformers capture these inter-modality interactions by repeatedly reinforcing

the characteristics of a target modality X_α with those of another source modality X_β .

4.3.2 Details of the proposed method

In this section we explain the different components of our proposed model (see Fig. 4.1).

1D Convolution Layer:

Following the same architecture as MulT, we use a 1D convolution layer over the temporal dimension to project the features of the different modalities into the same dimension and also to capture the dependencies between each element of the input sequences and its neighboring elements (local attention).

Crossmodal Transformer with Gated Residual Network:

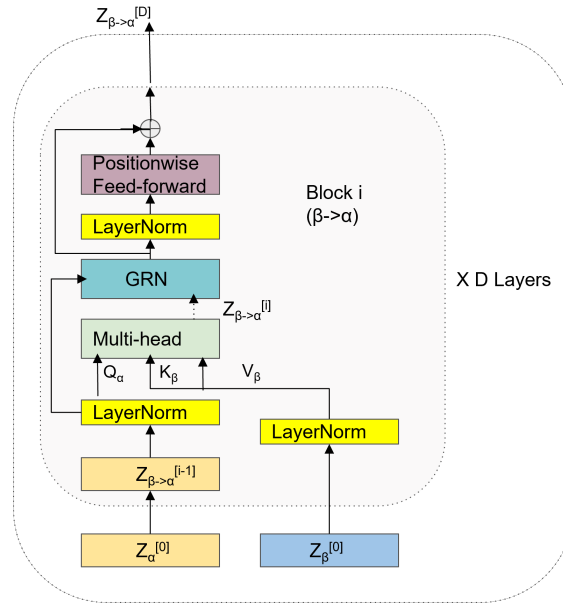


Figure 4.2 Crossmodal Transformer Block between two modalities α and β , where α is the target modality and β is the source modality. The main components of the crossmodal transformer are the co-attention operation to calculate the enhanced features for the target modality, and the Gated Residual Network to suppress unnecessary information coming from the source modality.

We aim to improve emotion recognition and sentiment analysis by adding a Gated Residual Network (GRN) [31] in multimodal transformers. We decided to add a GRN after the co-attention operation in each crossmodal transformer which will act as a filtering mechanism that will eliminate the unnecessary information calculated by the co-attention operation.

Our new crossmodal transformer block is shown in Fig. 4.2. In each crossmodal transformer, only two modalities are used: the target modality and the source modality. In our case, the target modality is always the language modality because linguistic input usually contains more information that can be used to extract emotions or sentiments. In addition, an unimodal transformer that uses textual data usually performs better than an unimodal transformer that uses acoustic or visual data. The crossmodal transformer module aims to enhance the features of the target modality with information coming from the source modality. The layers of the crossmodal transformer consist of multi-headed co-attention that take the feature sequences of two modalities and compute the attention between them. Instead of a simple residual connection, we added a Gated Residual Network that can skip over the multi-head co-attention block. This GRN will learn to combine the features of the target modality (the output from the previous block) and the enhanced features (the output of the co-attention). In this way, the crossmodal transformer will also be able to learn identity mapping. By using GRN, the crossmodal transformer will avoid unnecessary co-attention operations that are used in the architecture.

We selected the model of Tsai et al. [23] for our based multimodal transformer architecture. This model, named MulT, is mainly based on co-attention instead of self-attention in its layers. The co-attention mechanism aims to enhance the features of the target modality with those from the source modality. Given two modalities α and β , the feature sequences extracted from these modalities are denoted by $X_\alpha \in \mathbb{R}^{T_\alpha \times d_k}$ and $X_\beta \in \mathbb{R}^{T_\beta \times d_k}$ respectively, where T_α represents the sequence length and d_k represents the feature dimension.

Let us suppose that modality α is the target modality and β is the source modality. To compute the co-attention between these two modalities, the queries Q_α will be computed from the target modality and the keys and the values K_β, V_β from the source modality with

$$Q_\alpha = X_\alpha W_{Q_\alpha}, \quad (4.1)$$

$$K_\beta = X_\beta W_{K_\beta}, \quad (4.2)$$

and

$$V_\beta = X_\beta W_{V_\beta}, \quad (4.3)$$

where $W_{Q_\alpha} \in \mathbb{R}^{d_\alpha \times d_k}$, $W_{K_\beta} \in \mathbb{R}^{d_\beta \times d_k}$ and $W_{V_\beta} \in \mathbb{R}^{d_\beta \times d_v}$ are the weights. For the first block $X_\alpha = Z_\alpha^0$ which is the low-level feature of the target modality, and $X_\beta = Z_\beta^0$ which is the low-level features of the source modality. At each layer of the crossmodal transformer, the low-level features of the modality β are used as the source feature sequences. For an intermediate block i , $X_\alpha = Z_{\beta \rightarrow \alpha}^{i-1}$ and $X_\beta = Z_\beta^0$.

The co-attention from β to α is:

$$Y_\alpha = CM_{\beta \rightarrow \alpha}(X_\alpha, X_\beta) = \text{softmax}\left(\frac{Q_\alpha K_\beta^\top}{\sqrt{d_k}}\right)V_\beta. \quad (4.4)$$

For an intermediate block i , $Y_\alpha = Z_{\beta \rightarrow \alpha}^i$ which is the result of the co-attention operation.

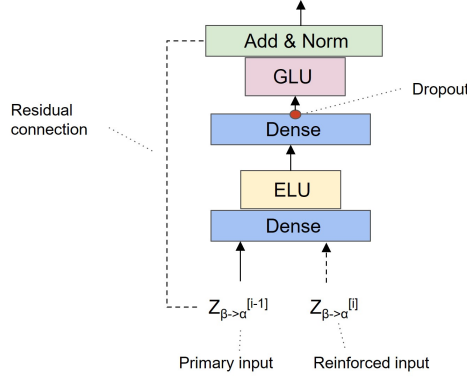


Figure 4.3 Gated Residual Network

The softmax calculates an attention matrix between the two modalities α and β . The output Y_α of co-attention represents the enhanced features that are going to be used as the target modality in the next layer. In addition, a residual connection and a feed-forward layer are added to the crossmodal transformer after each co-attention operation.

We modified this base model to include a GRN. This GRN takes a primary input (the features of the target modality, $Z_{\beta \rightarrow \alpha}^{i-1}$) and an optional context vector (in our case the context vector is $Z_{\beta \rightarrow \alpha}^i$, which is the vector resulting from the co-attention operation). Under the assumption that sometimes information sent from other modalities to the target modality may negatively affect the prediction instead of improving it, the GRN will learn to remove unnecessary information sent from other modalities. As shown in Fig. 4.3, the GRN will produce a vector that can either be the features of the target modality if the model depends only on that modality (the GRN will act as an identity function), or it can combine the features of the target modality with enhanced features that contain information from the source modality. The GRN will also give the crossmodal transformer the flexibility to apply non-linear processing only when necessary. The GRN generates its output as follows:

$$GRN_\omega(a, c) = LayerNorm(a + GLU_\omega(\eta_1)) \quad (4.5)$$

$$\eta_1 = W_{1,\omega}\eta_2 + b_{1,\omega} \quad (4.6)$$

$$\eta_2 = ELU(W_{2,\omega}a + W_{3,\omega}c + b_{2,\omega}) \quad (4.7)$$

$$GLU_\omega(\gamma) = \sigma(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega}) \quad (4.8)$$

where a is the primary input, c is the context vector, *LayerNorm* is a layer normalization, η_1 and η_2 are intermediate layers, $W_{\odot}$ and $b_{\odot}$ are the weights and biases respectively, *ELU* is the Exponential Linear Unit activation function, *GLU* is a Gated Linear Unit, and $\odot$ is an element-wise Hadamard product.

Gated Residual Networks can learn identity mapping more easily than residual connections. If the output of GLU is close to 0, then the entire co-attention module is removed, so feature enhancement will only be applied where it is needed. GRN will remove any co-attention operation from the architecture when it is unnecessary for the given input. If co-attention is applied, then GRN will learn how to combine the reinforced features with the output of the previous layer. In addition, GRNs perform better than residual connections in deeper networks [63].

Self-Attention Transformer:

Finally, the outputs of the crossmodal transformers are concatenated together. The resulting feature vector from the concatenation is passed through a self-attention transformer to capture temporal information. Finally, the output of the transformer is passed through a fully connected layer for the prediction.

The loss functions that were used depend on the dataset. For the IEMOCAP dataset, we used the Cross Entropy Loss between the inputs and the targets. For CMU-MOSI and CMU-MOSEI datasets we used the L1 Loss function which minimizes the sum of the absolute differences between the inputs and the targets.

4.4 Experiments

4.4.1 Datasets and evaluation metrics

To evaluate our proposed model, three datasets were used: CMU-MOSI [22], CMU-MOSEI [60] and IEMOCAP [64]. CMU-MOSI is used for multimodal sentiment intensity and subjectivity analysis. It consists of 2,199 online video clips annotated with labels for subjectivity and sentiment intensity at the opinion level. CMU-MOSEI dataset is considered as the largest dataset for multimodal emotion recognition and sentiment analysis. This dataset consists of 23,500 sentence utterance video clips from YouTube. CMU-MOSEI is annotated for both sentiment analysis and emotion recognition tasks. Finally, IEMOCAP consists of approximately 12 hours of audiovisual data. The

dataset contains multiple sessions where in each session two speakers are performing improvisations or scripted scenarios. IEMOCAP is annotated using 9 emotions but only 4 of them are selected for emotion recognition(happy, sad, angry, neutral).

For CMU-MOSI and CMU-MOSEI datasets, the evaluations metrics that are used are: the 7-class accuracy used for the sentiment score classification (acc7), the 5-class accuracy (acc5), the 2-class accuracy (Accuracy), F1 score (F1), mean absolute error (MAE) and the correlation (CC) of model’s prediction with true labels. For the IEMOCAP dataset, the binary classification accuracy and the F1 score are used to evaluate the model.

4.4.2 Implementation details

Features

We trained and evaluated our model using the extracted features from the work of MulT [23]. For the language modality, features were extracted from text using Glove [65] pre-trained for word embedding. The size of an embedding is 300. For the visual modality, Facet [66] was used to extract 35 facial action units from each frame. COVEREP [67] was used to extract features for the acoustic modality.

Hyperparameters

Table 4.1 displays the various hyperparameters employed in training our model. To ensure a fair comparison between MulT and our proposed method, the same hyperparameters, except for the additional one for the Gated Residual Network (GRN), were utilized to re-train MulT from scratch. This approach aimed to maintain consistency in the comparison.

Table 4.1 Hyperparameters

Batch Size	24
Number of Epochs	40
Initial Learning Rate	1e-3
Number of Layers	5
Optimizer	Adam
Crossmodal Attention Heads	5
Dropout for Crossmodal Attention Block	0,1
Textual Embedding Dropout	0,25
Dropout for the GRN	0,1
Output Dropout	0,1

MulT, when trained with the specified hyperparameters, demonstrated results on various metrics

that closely aligned with those obtained using Tsai’s proposed hyperparameters for different benchmark datasets.

For CMU-MOSI, CMU-MOSEI, and IEMOCAP, both MTGR and MulT reached convergence after 40 epochs in our experiments.

It is noteworthy that the remaining state-of-the-art models were not subjected to re-training in our study.

4.4.3 Results and discussion

In this section, we provide the results of our proposed model (MTGR) and compare these results with the results for state-of-the-art methods on the three datasets. Table 4.2 gives the results of our method on the CMU-MOSEI dataset. Table 4.3 gives the results of our method on the CMU-MOSI dataset. Table 4.4 gives the results of our method on the IEMOCAP dataset.

We recall that our proposed method and MulT were trained in the same way. The results show the effectiveness of the GRN as a filtering mechanism that is able to suppress unnecessary information. If the GRN is always acting like an identity function, it means that the information coming from the source modality is crucial and helps to improve the prediction results.

For the CMU-MOSI and IEMOCAP datasets, the results show that our method performs better than the original Multimodal Transformer. It succeeded to improve all the different metrics that are used for evaluation. For the CMU-MOSEI dataset, we can only see some improvements in the 7-class accuracy and the 5-class accuracy metrics.

We also compare our proposed method with several state-of-the-art models on the three benchmark datasets. For CMU-MOSI and CMU-MOSEI datasets, the results on the different metrics show that our method is very competitive and sometimes outperforms state-of-the-art models. For the IEMOCAP dataset, we noticed that our method was able to improve the results of the least present emotions in the dataset (Neutral, Sad, and Angry), which shows that our method is also robust to the data imbalance problem compared to other SOTA methods.

Table 4.2 Results on the CMU-MOSEI Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MulT [23]	0,614	0,673	0,493	0,509	0,810	0,808
GraphMFN [60]	0,71	0,54	0,45	-	0,77	0,769
RAVEN [61]	0,614	0,662	0,50	-	0,79	0,79
Transformer-based joint-encoding [68]	-	-	0,45	-	-	0,82
MTGR	0,617	0,657	0,497	0,513	0,794	0,795

Table 4.3 Results on the CMU-MOSI Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MuT [23]	1,030	0,657	0,335	0,384	0,773	0,774
RAVEN [61]	0,915	0,691	0,332	-	0,766	0,78
MTGR	0,904	0,691	0,367	0,413	0,803	0,803

Table 4.4 Results on the IEMOCAP Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	Neutral		Happy		Sad		Angry	
	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MuT [23]	0,690	0,692	0,836	0,859	0,826	0,835	0,851	0,845
RAVEN [61]	0,693	0,697	0,858	0,873	0,831	0,834	0,8671	0,873
MTGR	0,718	0,723	0,841	0,865	0,848	0,851	0,873	0,869

4.5 Conclusion

We proposed an improvement to the multimodal transformer (MuT) [23] for sentiment analysis and emotion recognition by adding a Gated Residual Network (GRN) [31] to the original end-to-end architecture. This gives the model more flexibility to use or suppress information received from other modalities. Our method showed an improvement in both F1 score and accuracy metrics for the CMU-MOSI and IEMOCAP datasets compared to MuT. Our method also can compare with, and sometimes outperforms, the state-of-the-art models.

CHAPTER 5 MULTIMODAL PRE-TRAINED TRANSFORMERS FOR EMOTION RECOGNITION AND SENTIMENT ANALYSIS

Transformers have achieved great success in many domains. Unlike other sequence models, transformers have the capability to handle and process long sequences of data in parallel. They have been leading the state-of-the-art of nearly all NLP tasks. After discovering transfer learning approaches, it has been a common practice to pre-train the transformer layers on different tasks before fine-tuning them on a downstream task. This practice has been widely used in language modelling problems where the transformer is first trained on large amounts of textual data and on various language modelling tasks, such as masked language modelling, next sentence prediction, etc. Training the model on these different problems helps to develop universal language knowledge. Pre-trained transformers are then fine-tuned on a new specific task.

Most research on transfer learning has traditionally proposed to refine a model on a downstream task that is similar to the upstream tasks. Furthermore, most of the proposed transformer-based pre-trained models have been limited to using downstream tasks with the same input modality as the dataset used in the pre-training process. For example, Rietzler et al. [69] used a pre-trained BERT architecture, first fine-tuning it in a self-supervised way on domain-specific corpora for the masked language model objective and next-sentence prediction tasks, then further fine-tuning the model on a domain-specific task for sentiment classification using text modality as input data. Other researchers have explored transfer learning with image-based datasets. However, a common approach involves ensuring that the modality used in the pre-training phase matches the modality used in the subsequent fine-tuning phase. For instance, Junlong et al. in their work [70] introduced DiT, a self-supervised pre-trained document image transformer. They pre-trained the proposed transformer on the masked image modeling task and subsequently fine-tuned it across a range of document AI tasks. Similarly, Minghao et al. in their work [71] presented a transformer-based pre-trained model designed for document digitization. Their model underwent pre-training and fine-tuning on text recognition tasks, employing textline images as inputs.

Recently, Lu et al. [72] have researched the ability of natural language pre-trained transformers to generalize to other data modalities. Inspired by their work and other multimodal transformer architectures, we propose and compare various multimodal pre-trained transformers that primarily consist of two key components:

- a fusion mechanism designed to effectively combine information from diverse modalities;
- and a pre-trained transformer model, specifically GPT-2 [73], tasked with extracting complex patterns and enhancing the representativeness of features derived from the different data modalities.

We refine the proposed architectures on downstream emotion recognition and sentiment analysis tasks. The pre-trained transformer model and fusion mechanisms are described in the following.

5.1 Pre-trained GPT-2 transformer model

GPT-2 is a transformer-based deep neural network with 1.5 billion parameters. GPT-2 architecture consists of a stack of transformer-based decoder blocks. Each decoder block comprises a self-attention layer, a feedforward neural network, and a normalization layer.

Figure 5.1 gives an overview of the GPT-2 architecture.

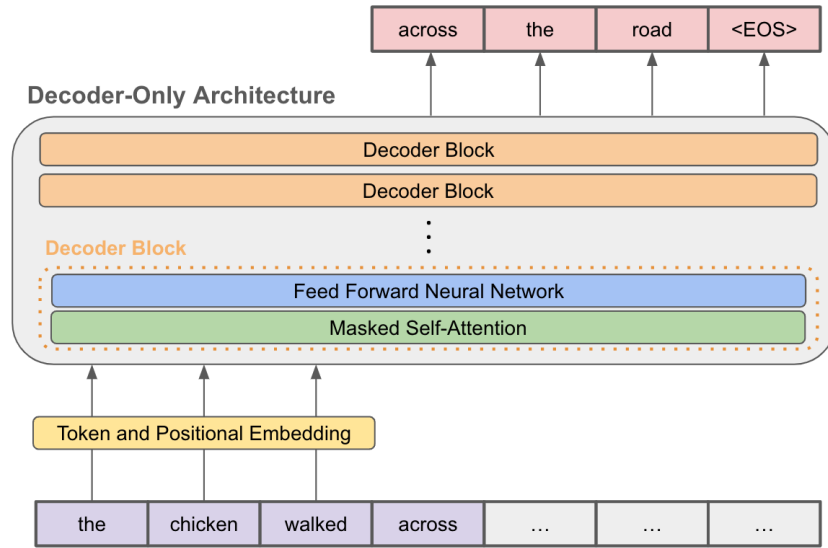


Figure 5.1 GPT-2 architecture is mainly composed of positional embeddings combined with token embeddings, and decoder blocks. Each decoder block comprises a masked self-attention layer and a feedforward layer [10] © Cameron R. Wolfe, 2022.

The model that we use has been pre-trained on 7,000 books and trained on a dataset of 8 million web pages to acquire an understanding of the language that can then be effectively fine-tuned on specific tasks.

5.2 Fusion mechanisms

Figure 5.2 provides an overview of the proposed architectures. Some of these architectures have been designed and tested on dual data modality inputs, while others are specifically configured and tested on triple data modality inputs. In the context of Figure 5.2, considering that a model incorporates three data modalities as inputs, X , Y , and Z correspond to the distinct data modalities: text, image, and audio. They are fed into a deep learning model. This deep learning model will learn

to combine and extract a vector of high-level features. Subsequently, these high-level features are utilized within a classification network to detect emotions or sentiments.

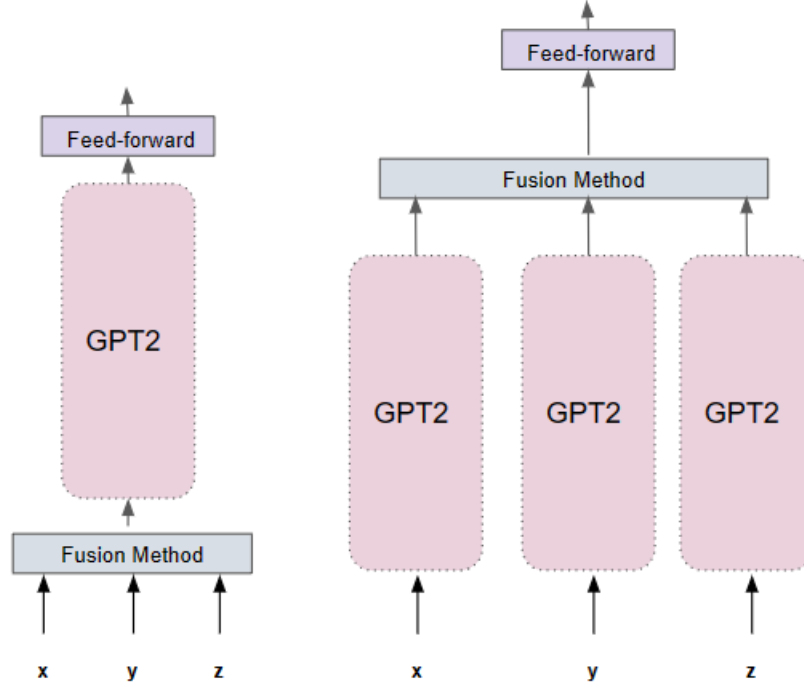


Figure 5.2 Overview of multimodal pre-trained transformer architectures is presented here. The initial architecture involves the fusion of low-level features extracted from diverse data modalities. The resultant features are subsequently inputted into a pre-trained transformer, such as GPT-2, to extract high-level features. These high-level features are then employed by a feedforward network to predict emotions. In contrast, the second architecture takes a different approach. It first extracts high-level features independently from each modality using GPT-2 transformers. Subsequently, the outputs of these GPT-2 models are fused, and the resulting feature vector is fed into a feedforward network to predict emotions.

The fusion of information can be performed on low-level or high-level features, depending on the model architecture. In addition, one or more pre-trained GPT-2 models can be used, according to the proposed fusion strategy and the number of input modalities. The fusion strategy plays a key role in our pre-trained multimodal emotion recognition architectures. Inspired by our work on MTGR in Chapter 4 and multimodal transformers, we propose several ways of fusing different data modalities. We also investigate the ability of pre-trained language modeling transformers (GPT-2) to generalize to multimodal data. In the following, we describe the different architectures compared in our work.

We first apply a 1D convolution layer to all the input features extracted from the different data modalities. This layer projects the features of the different modalities into the same dimension.

5.2.1 Architecture A: Concatenation

The first proposed architecture (see Figure 5.3) is a simple, shallow architecture based on the concatenation of input data modalities. The feature representations extracted from the *1D Conv* layers are concatenated together into a single feature vector. Next, the resulting feature vector is fed into the pre-trained GPT-2 to extract high-level features. Finally, the GPT-2 token *CLS* is fed into a feedforward neural network to classify emotions or sentiments.

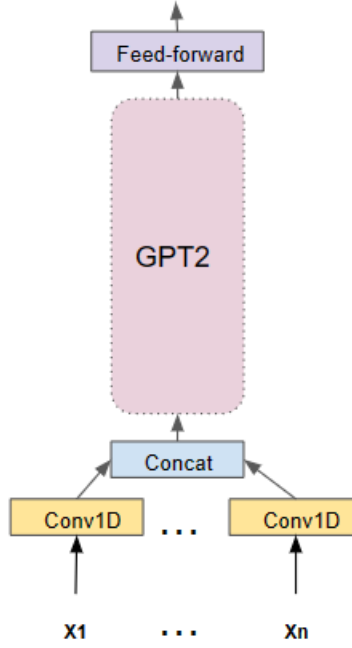


Figure 5.3 The architecture featuring GPT-2 with concatenation as the fusion method. The initial step involves the concatenation of the modalities X_1 to X_n , forming a combined input, which is then fed into a pre-trained GPT-2 model for the extraction of high-level features. Subsequently, the resulting vector is passed into a feedforward network for making predictions.

5.2.2 Architecture B: GRN and high-level features

In this architecture, we employ GRNs as our fusion modules. Figure 5.4 shows the details of this architecture. Each GRN takes two inputs: a primary input modality and an optional secondary input modality (context vector). First, the features projected from each modality by the *1D Conv* layers, are simultaneously fed as input to pre-trained GPT-2 models for the extraction of high-level feature representations. Subsequently, we introduce the GRN, which possesses the flexibility to learn whether the output prediction depends solely on the primary modality or if a combination with the secondary modality is necessary to enhance the accuracy of predictions.

In the case of n modality inputs, we employ an $(n - 1)$ -level GRN architecture. The first GRN takes as input the *CLS* tokens from the first two GPT-2 models, designating one *CLS* token as the primary input and the other as the secondary input. The subsequent GRN takes the resulting vector from the first GRN layer and the *CLS* token from the third GPT-2 model, with the output of the first GRN serving as the primary input and the *CLS* token as the secondary input, and so forth. Ultimately, the feedforward network will receive, as its input, the output generated by the final GRN layer in order to make predictions.

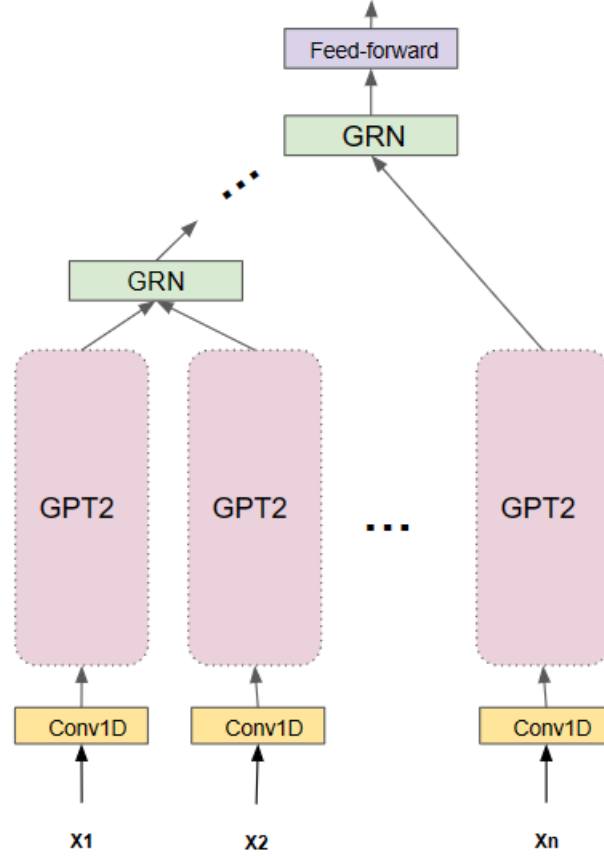


Figure 5.4 The architecture featuring GPT-2 with GRN serving as the fusion method for high-level features. The process begins by extracting high-level features utilizing pre-trained GPT-2 models. Following this, GRN is employed to combine the outputs from the GPT-2 models. Finally, the resultant vector is fed into a feedforward network to make predictions.

5.2.3 Architecture C: GRN and low-level features

This architecture closely resembles the previous one, with the primary distinction being the role of the GRN, which is now responsible for merging the low-level features. Figure 5.5 shows the details of this architecture. Initially, the projected feature vectors from two modalities of data are inputted

to the GRN, which has the capacity to either combine these two feature vectors or, alternatively, return the low-level features of the primary modality. In situations involving n modality inputs, we can implement an $(n - 1)$ -level GRN architecture. The initial GRN layer accepts inputs from two data modalities, distinguishing one as the primary input and the other as the secondary input. Subsequently, the following GRN layer receives the resultant vector from the initial GRN and introduces a new modality as the secondary input, with the output of the first GRN serving as the primary input and the new modality as the secondary input, and so on. Finally, the output of the last GRN layer is fed into the GPT-2 model to extract the high-level features that can be used for classification.

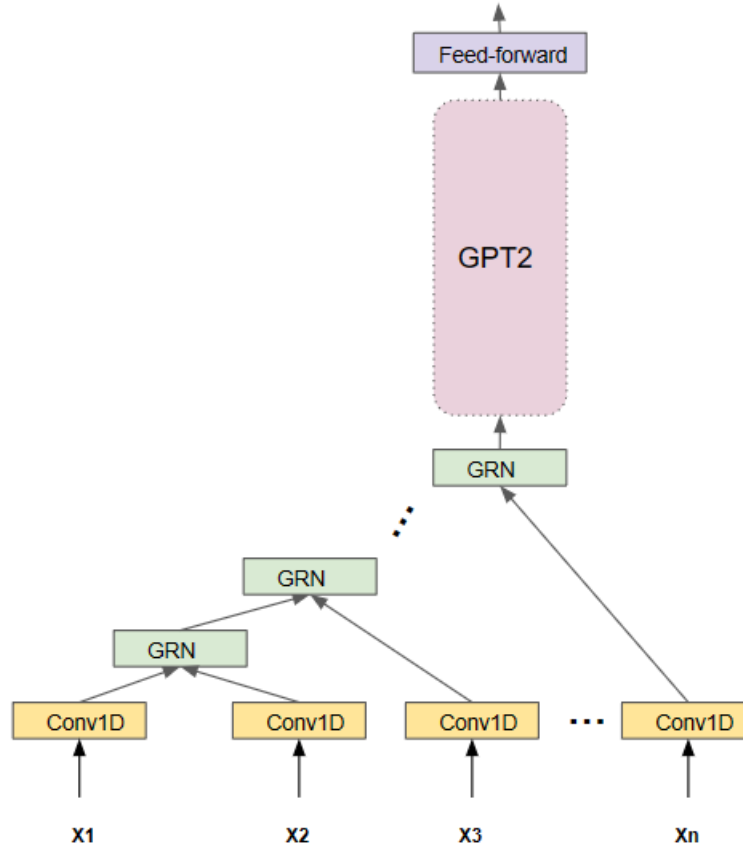


Figure 5.5 The architecture featuring GPT-2 with GRN serving as the fusion method for low-level features. The process begins by using GRNs to amalgamate the projected input modalities. The resulting fused vector is subsequently input into a pre-trained GPT-2 to extract a high-level feature vector, which is then utilized by a feedforward network for predicting results.

5.2.4 Architecture D: Co-attention and low-level features

In this architecture, we introduce a co-attention layer, aimed at merging feature vectors from distinct modalities. This architecture is illustrated in Figure 5.6.

As presented in the previous chapter (Chapter 4), co-attention involves the calculation of attention between modalities, specifically the target modality and source modalities. The fundamental objective of co-attention is to enhance the characteristics of the target modality by integrating features from the source modalities.

In this architecture, after the $1D$ *Conv* layers, we introduce a co-attention module to reinforce the characteristics of the target modality by leveraging those of the source modality. In scenarios involving n modality inputs, we employ an $(n - 1)$ -level co-attention architecture. The initial co-attention layer receives as input the projected feature vectors from the first two modalities, designating one as the target modality and the other as the source modality. Subsequently, the second co-attention layer takes the output from the first co-attention as the new target modality and the input from the new modality as the source modality, and so forth. The enhanced feature vector resulting from the last co-attention module is then directed into the GPT-2 model. Finally, consistent with previous architectures, the GPT-2 model output is fed into a feedforward network for the purpose of predicting emotions or sentiments.

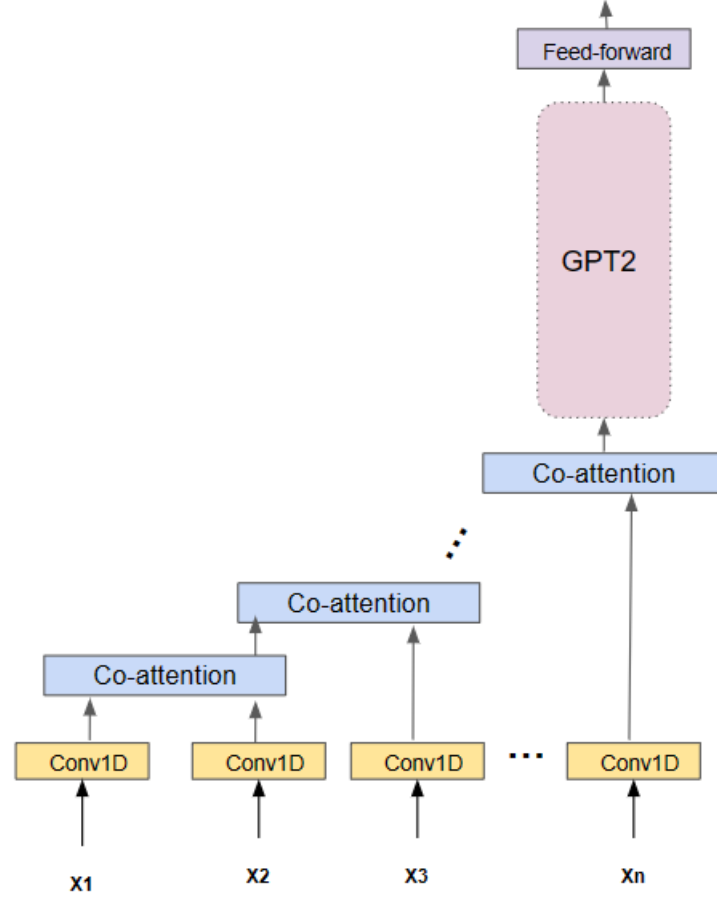


Figure 5.6 The architecture featuring GPT-2 with co-attention layers as the fusion method for low-level features. The process begins by using co-attention layers to amalgamate the projected input modalities. The resulting fused vector is subsequently input into a pre-trained GPT-2 to extract a high-level feature vector, which is then utilized by a feedforward network for predicting results.

5.2.5 Architecture E: Co-attention for low-level features and GRN for high-level features

Figure 5.7 shows the details of this architecture. Two co-attention layers are used in this architecture: the first layer enhances the features of modality X by the features of modality Y , and the second co-attention layer enhances the features of modality Y by the features of modality X . The co-attention layers will produce enhanced features for both modalities X and Y . Each enhanced feature vector is fed into its corresponding GPT-2 model. Finally, a GRN layer is added at the end to combine the high-level enhanced feature vectors. The output of the GRN layer is fed into a feedforward network for prediction. This architecture was applied only in the case of two modalities.

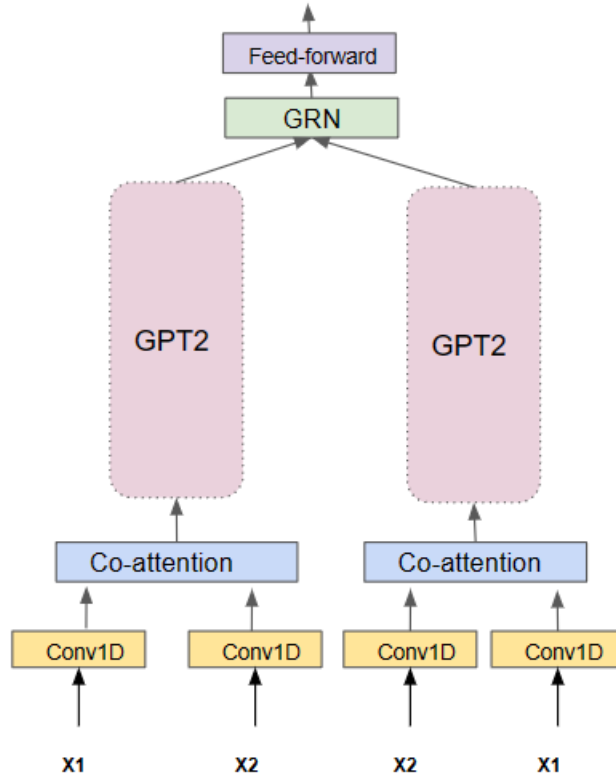


Figure 5.7 The architecture featuring GPT-2 with co-attention layers as the fusion method for low-level features and GRN for high-level features. The process begins by using co-attention layers to merge two distinct modalities. The resultant enhanced feature vectors are then input into a pre-trained GPT-2 to derive high-level feature vectors. Following this, GRN is employed to combine the high-level features. Finally, the resulting vector from GRN is forwarded into a feedforward network for prediction.

5.2.6 Architecture F: Co-attention and GRN for high-level features

Figure 5.8 shows the details of this architecture. This architecture is similar to that described above (Architecture E), but instead of applying co-attention between low-level features, we will apply it to the outputs of GPT-2 models in order to enhance high-level features. The outputs of the co-attention layers are then merged using the GRN layer. Finally, the resulting GRN vector is fed into the feedforward network.

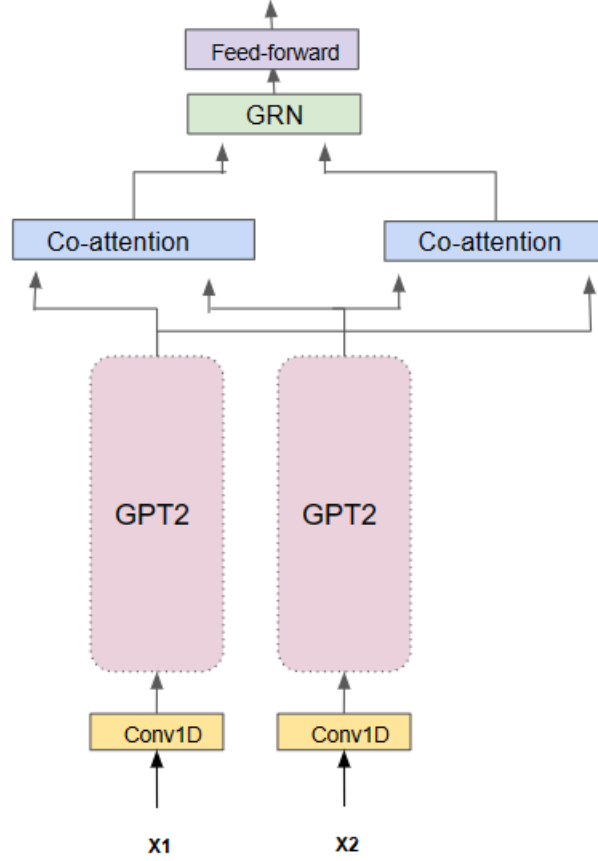


Figure 5.8 The architecture featuring GPT-2 along with co-attention layers and GRNs for fusing high-level features. Initially, GPT-2 extracts high-level feature vectors. Subsequently, co-attention layers enhance these high-level feature vectors, and the improved features are combined using a GRN layer. Following this, the resulting vector from the GRN is fed into a feedforward network for making predictions.

This architecture was applied only in the case of two modalities.

5.3 Fine-tuning

In line with the work of Lu et al. [72], we use the same fine-tuning method of freezing the self-attention and feedforward layers of the pre-trained GPT-2 model and fine-tuning the other layers. Their studies demonstrated that, without fine-tuning the self-attention and feedforward layers, their frozen pre-trained transformer (FPT) achieved better results than fully training a transformer from scratch for downstream tasks. In addition, the FPT performed better than a fully fine-tuned pre-trained transformer. They further demonstrated that the FPT can be adapted to completely different modalities. They explained the reason for the successful results obtained using FPT and suggested that the self-attention layers learned from linguistic models have acquired certain

properties that enable them to perform universal computation.

In our case, some layers will be trained from scratch, including 1D convolution layers, GRN layers, co-attention layers, and the feedforward network.

5.4 Experiments

To train and evaluate the proposed architectures, we used the same implementation details as those presented in the previous chapter (MTGR). The only difference is that we have replaced the transformer blocks with the pre-trained GPT-2 model. We have retained the same hyperparameters used for the GRN, co-attention, and feedforward layers as in the MTGR paper (Chapter 4).

Table 5.1 presents the various modality inputs that were tested for each architecture.

Table 5.1 Architectures/models and their modality inputs

Architecture A	Two modalities: Text and Audio
Architecture B	Two modalities: Text and Audio; Three modalities: Text, Audio, and Image
Architecture C	Two modalities: Text and Audio
Architecture D	Two modalities: Text and Audio
Architecture E	Two modalities: Text and Audio
Architecture F	Two modalities: Text and Audio

For models based on two data modalities, we designate the features of the text modality as the primary input for the GRN module, and as the target input for the co-attention module. Meanwhile, the audio modality is assigned the role of being the secondary modality for the GRN module and the source modality for the co-attention module.

5.4.1 Comparison of the proposed architectures

Table 5.2 gives the results of our method on the CMU-MOSEI dataset [60], Table 5.3 gives the results of our method on the CMU-MOSI dataset [22], and Table 5.4 gives the results of our method on the IEMOCAP dataset [64]. The results show that the performances of the different models are very close to each other. However, some models perform slightly better than others.

For the CMU-MOSEI dataset, the results show that the use of three pre-trained GPT-2s and two stacked GRNs (Architecture B) outperformed the other proposed architectures. It succeeded in improving all the different measures used for evaluation, with the exception of the F1 score. For the CMU-MOSI dataset, the shallow architecture (the concatenation-based model: Architecture A) outperformed the other proposed models on all the evaluation measures. For the IEMOCAP dataset, the results are more mixed. The best-performing architecture for the Neutral and Anger emotions is the architecture composed of two co-attention layers for low-level feature fusion, two

pre-trained GPT-2 models, and a GRN layer for high-level feature fusion (Architecture E). For the Happy emotion, the architecture based on a single co-attention layer for to fuse low-level features (Architecture D) performed slightly better than the others. For the Sad emotion, the second architecture with two input modalities (Architecture B) has the best F1 score, however, architecture B when using the three different data modalities has the best accuracy.

Table 5.2 Results on the CMU-MOSEI Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
A	0,819	0,233	0,41	0,41	0,72	0,649
B	0,820	0,207	0,416	0,416	0,676	0,637
C	0,837	0,118	0,413	0,413	0,771	0,628
B (3 modalities)	0,808	0,248	0,422	0,422	0,707	0,666
D	0,828	0,157	0,415	0,415	0,741	0,640
E	0,819	0,197	0,414	0,414	0,701	0,642
F	0,837	0,024	0,413	0,413	0,771	0,662

Table 5.3 Results on the CMU-MOSI Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
A	1,276	0,338	0,260	0,276	0,623	0,625
B	1,320	0,289	0,231	0,239	0,596	0,599
C	1,320	0,276	0,218	0,223	0,593	0,592
B (3 modalities)	1,50	0,021	0,154	0,154	0,593	0,422
D	1,418	0,300	0,230	0,249	0,586	0,571
E	1,370	0,322	0,246	0,265	0,587	0,574
F	1,395	0,325	0,256	0,276	0,584	0,560

Table 5.4 Results on the IEMOCAP Dataset. **Boldface**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	Neutral		Happy		Sad		Angry	
	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
A	0,618	0,637	0,811	0,851	0,769	0,809	0,837	0,831
B	0,612	0,643	0,789	0,856	0,803	0,798	0,841	0,843
C	0,556	0,590	0,789	0,856	0,785	0,807	0,836	0,834
B (3 modalities)	0,523	0,614	0,789	0,856	0,788	0,815	0,834	0,839
D	0,568	0,614	0,813	0,856	0,736	0,793	0,752	0,796
E	0,656	0,657	0,793	0,856	0,8	0,807	0,846	0,855
F	0,640	0,651	0,789	0,856	0,795	0,809	0,836	0,834

5.4.2 Comparison with the state-of-the-art

We also conducted a comparative analysis by contrasting the results of our proposed architectures with the results of state-of-the-art methods, including the MTGR model introduced in Chapter 4. This evaluation is performed on three datasets: CMU-MOSEI, CMU-MOSI, and IEMOCAP.

Table 5.5, Table 5.6, and Table 5.7 present the results obtained by the top-performing architectures, as determined by their performance across various metrics. Additionally, we provide insights into the performance of state-of-the-art models on these three benchmark datasets (CMU-MOSEI, CMU-MOSI, and IEMOCAP), as well as the results achieved by our prior approach, MTGR (as discussed in Chapter 4).

Upon examination of the results for CMU-MOSEI and CMU-MOSI, it is apparent that the models based on pre-trained transformers did not yield superior performance compared to other state-of-the-art methods. Notably, the MTGR method continues to demonstrate the best state-of-the-art results, particularly on the CMU-MOSI dataset. For the IEMOCAP dataset, the results indicate that models based on pre-trained transformers were able to achieve performances that are reasonably competitive with state-of-the-art methods and MTGR. Notably, this holds true for several architectures: B (2 modalities) regarding the F1 score for the Sad emotion, B (3 modalities) for the Accuracy of the Sad emotion, D for the Happy emotion, and E for the Neutral and Angry emotions.

We believe these results can be attributed to the dataset complexity. Unlike CMU-MOSI and CMU-MOSEI, which draw data from various sources, IEMOCAP is relatively smaller and less diverse. This smaller size and reduced diversity may contribute to our models converging more easily, although it could pose challenges for generalization.

Furthermore, the annotations used across these datasets differ. CMU-MOSEI employs binary or three-class annotations (positive, negative, neutral), and CMU-MOSI provides sentiment intensity scores. In contrast, IEMOCAP features extensive annotation, it is annotated into multiple emotion

classes. This rich annotation can enable models to capture distinct patterns associated with each emotion type.

Table 5.5 Results on CMU-MOSEI Dataset in Comparison to State-of-the-Art Models. **Bold-face**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MuT [23]	0,614	0,673	0,493	0,509	0,810	0,808
GraphMFN [60]	0,71	0,54	0,45	-	0,77	0,769
RAVEN [61]	0,614	0,662	0,50	-	0,79	0,79
Transformer-based joint-encoding [68]	-	-	0,45	-	-	0,82
MTGR	0,617	0,657	0,497	0,513	0,794	0,795
C	0,837	0,118	0,413	0,413	0,771	0,628
B (3 modalities)	0,808	0,248	0,422	0,422	0,707	0,666
F	0,837	0,024	0,413	0,413	0,771	0,662

Table 5.6 Results on CMU-MOSI Dataset in Comparison to State-of-the-Art Models. **Bold-face**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	MAE $\downarrow$	CC $\uparrow$	mult_acc_7 $\uparrow$	mult_acc_5 $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MuT [23]	1,030	0,657	0,335	0,384	0,773	0,774
RAVEN [61]	0,915	0,691	0,332	-	0,766	0,78
MTGR	0,904	0,691	0,367	0,413	0,803	0,803
A	1,276	0,338	0,260	0,276	0,623	0,625
F	1,395	0,325	0,256	0,276	0,584	0,560

Table 5.7 Results on IEMOCAP Dataset in Comparison to State-of-the-Art Models. **Bold-face**: best results. $\uparrow$: the higher value of the metrics means the better performance of the model. $\downarrow$: the lower value of the metrics means the better performance of the model.

	Neutral		Happy		Sad		Angry	
	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$	F1 score $\uparrow$	Accuracy $\uparrow$
MuT [23]	0,690	0,692	0,836	0,859	0,826	0,835	0,851	0,845
RAVEN [61]	0,693	0,697	0,858	0,873	0,831	0,834	0,8671	0,873
MTGR	0,718	0,723	0,841	0,865	0,848	0,851	0,873	0,869
B	0,612	0,643	0,789	0,856	0,803	0,798	0,841	0,843
B (3 modalities)	0,523	0,614	0,789	0,856	0,788	0,815	0,834	0,839
D	0,568	0,614	0,813	0,856	0,736	0,793	0,752	0,796
E	0,656	0,657	0,793	0,856	0,8	0,807	0,846	0,855

CHAPTER 6 ANALYSIS AND DISCUSSION

In this section, we present a detailed discussion about the models we have devised to address our problem. As previously mentioned, we’ve adopted two distinct methodologies for multimodal sentiment analysis and emotion recognition. The first approach entails enhancing a transformer-driven model with a GRN, which serves as a filtering mechanism. In this method, linguistic information is reinforced by audio and visual information, rather than combining them. This approach aims to facilitate the flow of relevant information while filtering out unnecessary elements to prevent confusion. The second approach involves the utilization of GPT-2-based pre-trained models that incorporate either early fusion or late fusion modules, which consist of a combination of co-attention and GRN layers. This results in architectures named as GPT-2-GRN-based or GPT-2-Co-attention-based models.

6.1 Custom transformer vs GPT-2

In our experiments, a custom transformer-based model outperforms the pre-trained models that use a language model for all modalities. Despite the reputation of a pre-trained GPT-2 when it comes to processing textual data alone, it seems to us that the addition of visual and audio components requires less advanced processing of textual information and, consequently, a less complex natural language model. A pre-trained GPT-2 is very effective in learning and exploiting textual information. We believe that it places less emphasis on audiovisual information when the learning process from textual information is at an advanced stage, i.e. the model jumps directly to a conclusion and explores less audiovisual information. A custom transformer, on the other hand, and because learning from text is still in its early stages, gives the opportunity to take audiovisual information into account at a higher rate and to enhance textual information without confusion or redundancy by using GRN in the co-attention process.

6.2 Analysis of the GPT-2-based models

In the GPT-2-based models, we used different architectures. We tried models based on 3 modalities and models based on 2 modalities. The reason for this was to test the effectiveness of the pre-trained models in handling different numbers of modalities. In each model, we tried the GRN-based information enhancement process for high-level features (enhancement takes place after the input features have been introduced into the pre-trained GPT-2s) and for low-level features (enhancement takes place on the initial representations of the pre-processed features). We applied a similar approach to the models based on co-attention. This modeling choice was made to see if we could be conclusive about the effectiveness of GPT-2 in preserving the information enhancement process.

In addition, we wanted to see whether or not the basic feature required further pre-processing. In our experiments, such a conclusion cannot be drawn independently of the dataset studied. Specifically, when dealing with complex datasets that demand extensive preprocessing of the initial input features due to their inadequate representation or when the features of different modalities exhibit correlations that can result in redundancy when low-level co-attention is applied, we believe it is preferable to use high-level features. Preprocessing the initial features can help mitigate redundancy and yield a more suitable representation. However, for more straightforward datasets where the basic features effectively represent the modalities, low-level features are generally preferable.

The aim of co-attention is to update a target modality with low-level external information from a source modality. We therefore believe that to better present and modulate the interactions between different modalities, both high-level and low-level features should be used. Performing a single layer of co-attention on low-level features will capture low-level interactions while performing co-attention on high-level features will capture high-level interactions. Therefore, integrating co-attention into pre-trained GPT-2 transformers instead of self-attention layers will better modulate interactions between modalities at different levels. We also believe that the key and value of the source modality should be updated at each level according to the results of the previous co-attention layer. This can help to learn which information is useful and can effectively improve the features of the target modality.

In the case of emotion recognition, we found that model performance is emotion-dependent. This may prove that not all emotions are processed in the same way. Some emotions are more complex than others and require further processing to be well represented. To this extent, we think that further research should be conducted to study the effect of correlation between modalities and whether it is preserved in deeper architectures. An emotion-based preprocessing could be studied, as it aligns with the hypothesis that emotion interpretability presents different levels of complexity.

Overall, we believe that a pre-trained model would have some sort of prejudice on the initial textual features. As it has already been pre-trained on linguistic modeling datasets, we feel that the model might focus solely on the target modality and may not capture the nuances of other modalities to predict results. This could degrade model performance and lead to poorer presentations than our transformer-GRN-based model. Therefore, a more effective method would be to come up with more effective fusion approach that enables to capture interactions at the different levels of the pre-trained model.

An alternative approach involves going beyond using models solely pre-trained on language modeling tasks; instead, we can enhance the pre-training process by introducing supervision from another modality of data. This approach allows the pre-trained deep learning model to acquire general knowledge from a specific modality of data while being supervised by another modality. Such a setup facilitates the model capacity to learn interactions and patterns that connect the two distinct modalities. We believe that such a model can offer greater flexibility in effectively learning patterns

from diverse data modalities.

Contrastive Language Image Pretraining (CLIP) [74], a notable example in this context, proposed a novel pre-training method that enables an encoder to be pre-trained by a specific modality while being supervised by data from another modality. In their research, they demonstrated that by pre-training an image encoder using CLIP on image inputs with natural language supervision, these encoders surpassed the performance of the best publicly available ImageNet [75] models in various image-based tasks. Their model operates on the principle of jointly training an image encoder and a text encoder to predict which image is paired with which text. This is achieved by increasing the cosine similarity of correct pairs (image embedding, text embedding) while minimizing incorrect pairings. Consequently, this results in the creation of two pre-trained encoders: an image encoder pre-trained with natural language supervision and a text encoder pre-trained with image supervision. In their experiments, Resnet50 and ViT were employed as image encoders, along with a transformer as the text encoder. To assess the effectiveness of their method, they conducted zero-shot transfer experiments to examine the model ability to generalize to unseen datasets. In comparison to other approaches that utilize zero-shot transfer or even few-shot techniques, their zero-shot CLIP model achieved competitive results.

This method can be an efficient means of enabling pre-trained language model transformers to acquire a general understanding of textual data and develop visual comprehension of the information encoded in text inputs. It would be interesting to apply it to emotion recognition.

CHAPTER 7 CONCLUSION

7.1 Summary of Works

In this thesis, we proposed and developed MTGR, a transformer-based multimodal emotion recognition network that relies on Gated Residual Networks (GRNs) to enhance the co-attention layer. The GRN module will selectively combine the reinforced features of the co-attention and features of the target modality. The proposed model achieves state-of-the-art results on CMU-MOSI, CMU-MOSEI, and IEMOCAP datasets for different evaluation metrics. The use of the GRN module shows its benefits as a filtering mechanism to remove irrelevant information coming from other modalities of data.

The thesis also explored different multimodal pre-trained transformer architectures and showed their ability to generalize on multimodal data for emotion recognition and sentiment analysis downstream tasks. Our results showed that providing the perfect fusion method to finetune a pre-trained multimodal transformer is still a difficult task, considering that a simple concatenation method outperformed the other proposed methods on the CMU-MOSI dataset for all evaluation metrics.

7.2 Future Research

As our work was based on pre-extracted features, we believe that the methods used for feature extraction might affect the predictions of our proposed models. We suggest that the results may be improved by carefully selecting the low-level feature extractors.

We also propose to integrate the co-attention module into the pre-trained transformer instead of providing an external fusion method. The idea is to freeze the pre-trained transformer self-attention weights and modify its input vectors so that it learns co-attention between different data modalities. We believe that by using this method, the pre-trained transformer will be fine-tuned to learn the co-attention operation and modulate the interactions between the different modalities.

Another potential approach is to pre-train our data modality encoders using the CLIP method. In doing so, we believe that the encoder for the text modality, for instance, will acquire knowledge from textual data while simultaneously considering the visual context of the text, due to the supervision by image inputs. By employing CLIP, we expect that our transformers will display less bias toward any specific target modality, allowing for more efficient fine-tuning on downstream tasks. Furthermore, we believe that a pre-trained text encoder supervised by image inputs is better suited for fine-tuning a downstream task involving image input modality, in comparison to a transformer that was pre-trained solely on language modeling tasks. Further research and investigation should be conducted.

REFERENCES

- [1] <https://towardsdatascience.com/perceptrons-the-first-neural-network-model-8b3ee4513757>.
- [2] <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [3] <https://tikz.net/conv2d/>.
- [4] <https://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/>.
- [5] <https://www.analyticsvidhya.com/blog/2021/06/recurrent-neural-networks-introduction-for-beginners/>.
- [6] <https://isheunesu48.medium.com/recurrent-neural-networks-f06e2c263d84>.
- [7] https://thorirmar.com/post/insight_into_lstm/.
- [8] <https://vaclavkosar.com/ml/transformers-self-attention-mechanism-simplified>.
- [9] <https://www.jeremyjordan.me/nn-learning-rate/>.
- [10] <https://towardsdatascience.com/language-models-gpt-and-gpt-2-8bdb9867c50a>.
- [11] L. Pepa, L. Spalazzi, M. Capecci, and M. G. Ceravolo, "Automatic emotion recognition in clinical scenario: a systematic review of methods," *IEEE Transactions on Affective Computing*, 2021.
- [12] C. D. Katsis, N. Katertsidis, G. Ganiatsas, and D. I. Fotiadis, "Toward emotion recognition in car-racing drivers: A biosignal processing approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 3, pp. 502–512, 2008.
- [13] K. Fukuda and Q. Jin, "Analyzing change on emotion scores of tweets before and after machine translation," in *Social Computing and Social Media: Design, User Experience and Impact: 14th International Conference, SCSM 2022, Held as Part of the 24th HCI International Conference, HCII 2022, Virtual Event, June 26–July 1, 2022, Proceedings, Part I*. Springer, 2022, pp. 294–308.
- [14] S. K. Kimovska and V. Cvetkoski, "The effect of emotions on translations performance," *Research in Language*, vol. 19, no. 2, pp. 169–186, 2021.
- [15] D. Panda, D. D. Chakladar, and T. Dasgupta, "Multimodal system for emotion recognition using eeg and customer review," in *Proceedings of the global ai congress 2019*. Springer, 2020, pp. 399–410.

- [16] K. P. Seng and L.-M. Ang, “Video analytics for customer emotion and satisfaction at contact centers,” *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 3, pp. 266–278, 2017.
- [17] L. F. Barrett, K. A. Lindquist, and M. Gendron, “Language as context for the perception of emotion,” *Trends in cognitive sciences*, vol. 11, no. 8, pp. 327–332, 2007.
- [18] L.-P. Morency, R. Mihalcea, and P. Doshi, “Towards Multimodal Sentiment Analysis: Harvesting Opinions from The Web,” in *International Conference on Multimodal Interfaces (ICMI 2011)*, Alicante, Spain, Nov. 2011.
- [19] V. Pérez-Rosas, R. Mihalcea, and L.-P. Morency, “Utterance-level multimodal sentiment analysis,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 973–982. [Online]. Available: <https://aclanthology.org/P13-1096>
- [20] S. Poria, I. Chaturvedi, E. Cambria, and A. Hussain, “Convolutional mkl based multimodal emotion recognition and sentiment analysis,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 439–448.
- [21] H. Wang, A. Meghawat, L. Morency, and E. P. Xing, “Select-additive learning: Improving cross-individual generalization in multimodal sentiment analysis,” *CoRR*, vol. abs/1609.05244, 2016. [Online]. Available: <http://arxiv.org/abs/1609.05244>
- [22] A. Zadeh, R. Zellers, E. Pincus, and L. Morency, “MOSI: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos,” *CoRR*, vol. abs/1606.06259, 2016. [Online]. Available: <http://arxiv.org/abs/1606.06259>
- [23] Y. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L. Morency, and R. Salakhutdinov, “Multimodal transformer for unaligned multimodal language sequences,” in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2019, pp. 6558–6569. [Online]. Available: <https://doi.org/10.18653/v1/p19-1656>
- [24] A. Kumar and J. Vepa, “Gated mechanism for attention based multi modal sentiment analysis,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4477–4481.
- [25] K. Yang, H. Xu, and K. Gao, “Cm-bert: Cross-modal bert for text-audio sentiment analysis,” in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 521–528.
- [26] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65, no. 6, p. 386, 1958.

- [27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [28] M. I. Jordan, “Serial order: A parallel distributed processing approach,” in *Advances in psychology*. Elsevier, 1997, vol. 121, pp. 471–495.
- [29] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [31] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” 2020.
- [32] C. Strapparava and R. Mihalcea, “Learning to identify emotions in text,” in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 1556–1560.
- [33] Y.-S. Seol, D.-J. Kim, and H.-W. Kim, “Emotion recognition from text using knowledge-based ann,” in *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, 2008, pp. 1569–1572.
- [34] R. C. Balabantaray, M. Mohammad, and N. Sharma, “Multi-class twitter emotion classification: A new approach,” *International Journal of Applied Information Systems*, vol. 4, no. 1, pp. 48–53, 2012.
- [35] M. Hasan, E. Rundensteiner, and E. Agu, “Emotex: Detecting emotions in twitter messages,” 2014.
- [36] A. Seyeditabari, S. Levens, C. D. Maestas, S. Shaikh, J. I. Walsh, W. Zadrozny, C. Danis, and O. P. Thompson, “Cross corpus emotion classification using survey data,” *This paper was presented at AISB*, 2017.
- [37] N. I. Tripto and M. E. Ali, “Detecting multilabel sentiment and emotions from bangla youtube comments,” in *2018 International Conference on Bangla Speech and Language Processing (ICB-SLP)*. IEEE, 2018, pp. 1–6.
- [38] E. Batbaatar, M. Li, and K. H. Ryu, “Semantic-emotion neural network for emotion recognition from text,” *IEEE access*, vol. 7, pp. 111 866–111 878, 2019.
- [39] W. Ying, R. Xiang, and Q. Lu, “Improving multi-label emotion classification by integrating both general and domain-specific knowledge,” in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 2019, pp. 316–321.

- [40] H. Alhuzali and S. Ananiadou, “Spanemo: Casting multi-label emotion classification as span-prediction,” *arXiv preprint arXiv:2101.10038*, 2021.
- [41] J. Y. R. Cornejo, H. Pedrini, and F. Flórez-Revuelta, “Facial expression recognition with occlusions based on geometric representation,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 20th Iberoamerican Congress, CIARP 2015, Montevideo, Uruguay, November 9-12, 2015, Proceedings 20*. Springer, 2015, pp. 263–270.
- [42] V. Gomathi, K. Ramar, and A. S. Jeevakumar, “Human facial expression recognition using manfis model,” *International Journal of Computer and Information Engineering*, vol. 3, no. 2, pp. 288–292, 2009.
- [43] A. R. Rivera, J. A. R. Castillo, and O. Chae, “Recognition of face expressions using local principal texture pattern,” in *2012 19th IEEE international conference on image processing*. IEEE, 2012, pp. 2609–2612.
- [44] C. Pramerdorfer and M. Kampel, “Facial expression recognition using convolutional neural networks: state of the art,” *arXiv preprint arXiv:1612.02903*, 2016.
- [45] F. Xue, Q. Wang, and G. Guo, “Transfer: Learning relation-aware facial expression representations with transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3601–3610.
- [46] Z. Wen, W. Lin, T. Wang, and G. Xu, “Distract your attention: Multi-head cross attention network for facial expression recognition. arxiv 2021,” *arXiv preprint arXiv:2109.07270*.
- [47] F. Xue, Q. Wang, Z. Tan, Z. Ma, and G. Guo, “Vision transformer with attentive pooling for robust facial expression recognition,” *IEEE Transactions on Affective Computing*, 2022.
- [48] C.-H. Wu and W.-B. Liang, “Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels,” *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 10–21, 2010.
- [49] V. Petrushin, “Emotion in speech: Recognition and application to call centers,” in *Proceedings of artificial neural networks in engineering*, vol. 710, 1999, p. 22.
- [50] G. Keren and B. Schuller, “Convolutional rnn: an enhanced model for extracting features from sequential data,” in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3412–3419.
- [51] S. Mirsamadi, E. Barsoum, and C. Zhang, “Automatic speech emotion recognition using recurrent neural networks with local attention,” in *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 2227–2231.

- [52] Y. Li, T. Zhao, T. Kawahara *et al.*, “Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning.” in *Interspeech*, 2019, pp. 2803–2807.
- [53] L.-P. Morency, R. Mihalcea, and P. Doshi, “Towards multimodal sentiment analysis: Harvesting opinions from the web,” in *Proceedings of the 13th international conference on multimodal interfaces*, 2011, pp. 169–176.
- [54] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, “Tensor fusion network for multimodal sentiment analysis,” *arXiv preprint arXiv:1707.07250*, 2017.
- [55] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency, “Context-dependent sentiment analysis in user-generated videos,” in *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 2017, pp. 873–883.
- [56] P. Ekman, “Universals and cultural differences in facial expressions of emotion.” in *Nebraska symposium on motivation*. University of Nebraska Press, 1971.
- [57] R. Plutchik, “A general psychoevolutionary theory of emotion,” in *Theories of emotion*. Elsevier, 1980, pp. 3–33.
- [58] S. Dobrišek, R. Gajšek, F. Mihelič, N. Pavešić, and V. Štruc, “Towards efficient multi-modal emotion recognition,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 53, 2013.
- [59] B. Li, C. Li, F. Duan, N. Zheng, and Q. Zhao, “TPFN: applying outer product along time to multimodal sentiment analysis fusion on incomplete data,” in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIV*, 2020, pp. 431–447. [Online]. Available: https://doi.org/10.1007/978-3-030-58586-0_26
- [60] A. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L. Morency, “Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 2236–2246. [Online]. Available: <https://aclanthology.org/P18-1208/>
- [61] Y. Wang, Y. Shen, Z. Liu, P. P. Liang, A. Zadeh, and L.-P. Morency, “Words can shift: Dynamically adjusting word representations using nonverbal behaviors,” in *AAAI*, 2019, pp. 7216–7223. [Online]. Available: <https://doi.org/10.1609/aaai.v33i01.33017216>
- [62] J. Delbrouck, N. Tits, M. Brousmiche, and S. Dupont, “A transformer-based joint-encoding for emotion recognition and sentiment analysis,” *CoRR*, vol. abs/2006.15955, 2020. [Online]. Available: <https://arxiv.org/abs/2006.15955>

- [63] P. Savarese and D. Figueiredo, “Residual gates: A simple mechanism for improved network optimization,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [64] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, “IEMOCAP: interactive emotional dyadic motion capture database,” *Lang. Resour. Evaluation*, vol. 42, no. 4, pp. 335–359, 2008. [Online]. Available: <https://doi.org/10.1007/s10579-008-9076-6>
- [65] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [66] “imotions.” <https://imotions.com/>, 2017.
- [67] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, “Covarep—a collaborative voice analysis repository for speech technologies,” in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014, pp. 960–964.
- [68] J.-B. Delbrouck, N. Tits, M. Brousmitche, and S. Dupont, “A transformer-based joint-encoding for emotion recognition and sentiment analysis,” *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, 2020. [Online]. Available: <http://dx.doi.org/10.18653/v1/2020.challengehml-1.1>
- [69] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, “Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification,” *arXiv preprint arXiv:1908.11860*, 2019.
- [70] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei, “Dit: Self-supervised pre-training for document image transformer,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 3530–3539.
- [71] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, “Trocr: Transformer-based optical character recognition with pre-trained models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 11, 2023, pp. 13 094–13 102.
- [72] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, “Pretrained transformers as universal computation engines,” *arXiv preprint arXiv:2103.05247*, vol. 1, 2021.
- [73] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners.”
- [74] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

- [75] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.