ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Chair for Mathematical Information Science
Prof. Dr. H. Bölcskei

Student Project in Information Technology and Electrical Engineering

Spring Semester 2019

Gustavo Cid Ornelas

Multimodal Emotion Recognition Using Lexico-Acoustic Language Descriptions

Supervisor: Dmytro Perekrestenko

June 2019

Abstract

The problem of emotion recognition is challenging and of utmost importance in the area of Human-Computer Interaction. Most of the research so far focused on the classification of emotions based on speech only. But, just like humans experience the world from a multitude of modalities, there is an arising direction of research aiming to incorporate the multimodal nature of human experience to deep learning models. We explore the problem of emotion recognition with deep learning, using speech and text in a completely end-to-end manner on the IEMOCAP dataset. Our model diverges from the classical multimodal fusion approaches and strives to explore to the fullest the interplay between intra-modality and inter-modality dynamics, with insights drawn from the problem of neural machine translation.

Contents

Notation

Vectors and matrices are written in bold font. The superscript $^\intercal$ denotes the transpose of a matrix or of a vector. The function $\log(\cdot)$ denotes the base 10 logarithm and $\tanh(\cdot)$ denotes the hyperbolic tangent.

Chapter 1

Introduction

1.1 The problem of emotion recognition

Emotions are ubiquitous and greatly influence human interactions. They are, thus, a fundamental component of human communication. If we aim to make human-computer interaction natural and smooth, we must focus on developing emotionally aware intelligence. A first step towards this ultimate goal is the development of robust and reliable emotion recognition systems.

Emotion recognition is a hard problem. It is hard because different individuals express emotions differently. Moreover, emotions are expressed through a multitude of modalities, from speech to body language cues [1].

1.2 Related work

A lot of the research on emotion recognition focused on the classification of emotions based on speech alone. Researchers explored classical machine learning approaches, such as Hidden Markov Models (HMMs) [2] and Support Vector Machines (SVMs) [3]. More recently, the classical algorithms were beaten by the use of deep neural networks.

The first step in most of the proposed emotion detection schemes based on speech is feature extraction. The objective is to extract a set of features from the raw audio file that are capable of representing the emotional state. Extracting the proper set of features is crucial and strongly influences the models' performance [1]. Even though there has been significant amount of work on hand-crafting these sets of features, there is no consensus about which features are more informative about emotions [4].

In [5] , there is a recommendation of two sets of features that should be considered for the task of emotion recognition from speech. These sets were constructed based on their theoretical significance, past success and representation of the physiological changes in the voice during the affective process. These parameters are frequency, energy/amplitude and spectral related. Additionally, some statistical aggregator functions are proposed to group these features.

Most of the previous work applying deep learning to emotion recognition from speech make use of a combination of these features, being Mel-frequency cepstral coefficients (MFCCs) and the fundamental frequency (F0) the most common among them [4, 6–8].

There are also researchers who explored the problem of emotion recognition from other sources, such as text [9], video [10, 11] and EEG brainwaves [12].

1.3 Multimodal approaches

Human experience is intrinsically multimodal [13]. We obtain information from a variety of sources and combine them to make sense of the world.

In the problem of emotion recognition from speech, if we make use of features extracted from audio only, for instance, we are neglecting the fact that words carry meaning and that meaning can play a central role in the emotion being expressed. A high pitch, for example, should be interpreted very differently if it is associated with a very positive or a very negative word. The idea of multimodal models in the emotion recognition setting is, thus, combining the information from multiple modalities in order to recognize the emotion with a higher confidence.

There are different possibilities related to how one could go about combining the modalities. The most straightforward ones are feature-level fusion, in which the features are combined before being fed to the model, and decision-level fusion, in which there is one model for each modality and they are then merged together to produce a single output [14]. A simple instance of feature-level fusion is concatenation of features from the different modalities. As for decision-level fusion, one could feed the outputs from both modalities to a fully connected layer, for example. These are by no means the only possibilities of combining different modalities. Other approaches that differ from these two simpler cases can be explored in more details in [13].

The challenge of fusion in multimodal emotion recognition can be seen as fully exploiting the interplay between intra-modality and inter-modality

dynamics [15]. Inter-modality dynamics are the interactions between the different modalities, for example, between speech and the spoken content. Intra-modality dynamics can be seen as the idiosyncratic elements that characterize a modality, as an example, consider the nature of speech, where the conventional sentence structure is often neglected and there are pauses, laughter, among other elements. On the one hand, with feature-level fusion, the intra-modality dynamics are lost. On the other, in decision-level fusion, the inter-modality dynamics are not explored to the fullest.

This work combines the modalities in a way that is inspired in the encoder-decoder architecture from the problem of neural machine translation (NMT). We build a model comprised of two parts, one for each modality. One part helps the other by generating representations that are meaningful for the task considered. By doing so, we believe that we explore more thoroughly the interplay between the different dynamics that are the key in the multimodal setting. We differ from previous work: first, because we do not make use of engineered features, our model is fully end-to-end; second, because we diverge from the basic paradigm of feature/decision-level fusion, whereas the models that set the state-of-the-art can be seen as instances of decision-level fusion [16–18].

Chapter 2

The IEMOCAP Dataset

2.1 Dataset overview

In this chapter, we introduce the dataset used to train and evaluate all of the models considered in this work. The corpora used is the Interactive Emotional Dyatic Motion Capture (IEMOCAP) dataset [19]. The IEMOCAP aims to overcome the limitations of most of the available emotion recognition datasets, which present significant shortcomings, such as focus on the acoustic speech channel, limited size, small number of subjects, among others.

The dataset is divided into 5 sessions with 2 subjects each, one male and one female. In 3 of the sessions, the actors are asked to perform scripts and in the remaining 2 sessions, they are asked to improvise hypothetical scenarios, all of which are designed to elicit specific emotions. In total, there are 10 actors recorded, adding up to approximately 12 hours of data.

Besides speech and transcriptions, there are also available videos and motion capture information, but this work focused solely on speech and transcriptions. After the recording phase, the sessions were manually segmented into utterances of variable length and each utterance was annotated by at least 3 human annotators.

In order to be consistent with previous work, we considered only the samples in which the majority of human annotators agreed on an emotional category [17,16]. Moreover, we focused on only 4 categories, namely *happy*, *sad*, *angry* and *neutral*. Also, for consistency with previous research work, we merged the *excitement* with the *happy* categories.

Therefore, our final dataset is comprised of 5,531 utterances, of which 1,636 are *happy*, 1,084 are *sad*, 1,103 are *angry* and 1,708 are *neutral*.

2.2 Audio data

The audio file for each utterance is available in the .wav format. These utterances contain great variability in length.

We decided to truncate all of the audio files at 150,000 samples, in order to reduce them to a more manageable size. Since the audio data is sampled at 16 kHz, this is equivalent to recognizing an emotion after listening to 9.375 seconds of an utterance. Additionally, as a preprocessing stage, we standardized all of the audio samples to have zero mean and unit variance. This was done to remove the influence of the different levels of loudness between speakers' voices.

2.3 Text data

There are transcriptions available for all of the samples. The dataset used contains a vocabulary size of 3,747, which includes punctuation symbols and the "_UNK_" token, used for unknown words. The maximum transcription length is 128. To use it with the model, we tokenized and indexed all of the transcriptions.

Chapter 3

Methodology

3.1 Introduction

The main contributions of this work are related to the multimodal model for emotion recognition with an architecture inspired in the encoder-decoder structure with attention used in neural machine translation (NMT). In this chapter, we first go through the fundamental aspects of the problem of NMT and then show how the ideas are directly transferred to the problem of emotion recognition. Finally, we propose the models investigated. All of the discussion on NMT in this chapter is based on [20] and [21].

3.2 Neural machine translation (NMT)

The problem of translation can be seen essentially as the problem of modelling the conditional probability distribution $p(\mathbf{y}|\mathbf{x})$ of translating a source sentence $(\mathbf{x_1}, \ldots, \mathbf{x_n})$ into a target sentence $(\mathbf{y_1}, \ldots, \mathbf{y_m})$.

One of the most popular neural machine translation architectures is based on an encoder and a decoder. The encoder has a source sentence as its input and is responsible for generating a representation for that sentence. The decoder, then, is responsible for producing one target word at a time based on the representation provided by the encoder.

For instance, one could choose one recurrent neural network (RNN) for the encoder and one for the decoder, though that is by no means the only possible choice. The step by step of such NMT model would be:

- Read the input sentence, which is a sequence of vectors, $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_n})$, into a vector $\mathbf{c}$, which is called context vector

- Through the encoding RNN, we would be computing the hidden state at time t as $\mathbf{h_t} = f(\mathbf{x_t}, \mathbf{h_{t-1}})$, where $f(\cdot)$ could be a LSTM
- The context vector is a nonlinear function of the encoder hidden states, $\mathbf{c} = q(\mathbf{h_1}, \ldots, \mathbf{h_n})$. Early work used $\mathbf{c} = \mathbf{h_n}$, i. e., the last hidden state of the encoder
- The decoder, then, predicts the next word given the context vector and all of the previously predicted words

As mentioned, early work on NMT used the context vector c only as an initializer of the decoder's hidden state. The intuition behind it is that the encoding RNN would find a good representation for the whole sentence and then, the translation would be done over this representation. Albeit this approach proved to work for shorter phrases, it turned out to be a significant bottleneck for longer ones.

Another approach, which yields in better results, is to build the context vector from all or from a subset of encoder's hidden states, which are consulted throughout the translation process. This is referred to as an attention mechanism.

In terms of the attention mechanism, one can split the approaches into two categories, namely global and local. The difference between them is based on whether the context vector is built based on all of the encoder's hidden states or just on a subset of them, in other words, if the attention is placed on the whole encoder or just on some of its units.

3.2.1 Global attention mechanism

In this approach, the context vector is defined as a linear combination of all of the encoder's hidden states. Thus, at each time step, the decoder has a different context vector.

What varies at each time step are the weights that are used to compute the context vector. We are interested in weighting more heavily the parts of the encoder that have more to do with the word we are translating at that moment and weight less the parts that have less to do with it. This process can be seen as an alignment: we are constantly aligning the current state of the decoder with the part of interest in the encoder's states.

In this model type, there is a variable-length alignment vector $\mathbf{a_t}$, whose length equals the number of time steps in the source side. This alignment vector contains the weights that are going to be used to build the context vector and it is derived by comparing the current state of the decoder $\mathbf{h_t}$ with each of the encoder states $\bar{\mathbf{h}}_\mathbf{s}$.

$$\mathbf{a_t}(s) = \text{align}(\mathbf{h_t}, \bar{\mathbf{h_s}}) = \frac{\exp(\text{score}(\mathbf{h_t}, \bar{\mathbf{h_s}}))}{\sum_{s'=1}^{n} \exp(\text{score}(\mathbf{h_t}, \bar{\mathbf{h_{s'}}}))} \tag{3.1}$$

The score is a function that should measure the relation between the current decoder's hidden state $\mathbf{h_t}$ with each encoder's hidden states $\bar{\mathbf{h_s}}$. There are different possibilities for this score function. For instance, one could use a single layer feedforward neural network, as in [21], the inner product between $\mathbf{h_t}$ and $\bar{\mathbf{h_s}}$, as in [20], among other possibilities.

$$\text{score}(\mathbf{h_t}, \bar{\mathbf{h_s}}) = \begin{cases} \mathbf{h_t}^\intercal \bar{\mathbf{h_s}} & \text{as in [20]} \\ \mathbf{v_a}^\intercal \tanh\left(\mathbf{W_{a1}}\mathbf{h_t} + \mathbf{W_{a2}}\bar{\mathbf{h_s}}\right) & \text{as in [21]} \end{cases} \tag{3.2}$$

Based on the alignment vectors $\mathbf{a_t}$, the context vector $\mathbf{c_t}$ for the decoder time step t is given by Equation 3.3.

$$\mathbf{c_t} = \sum_{s=1}^{n} \mathbf{a_t}(s)\bar{\mathbf{h_s}} \tag{3.3}$$

Where we assume that the encoder has n time steps (n words to be translated). The alignment model is trained jointly with the translation model.

3.2.2 Local attention mechanism

The global attention mechanism has the drawback that we have to look over the whole encoder to compute the alignment vector. This process can be expensive computationally when we have longer input sequences such as paragraphs or whole documents. Thus, to circumvent this drawback, we can focus on just a subset of the encoder's units, i. e., focus only in a small window of context.

At each decoder time step t, the model generates a an aligned position p_t . The context vector $\mathbf{c_t}$ is, then, computed as a weighted average over the units within the window $[p_t - D, p_t + D]$, where the window length $2D$ is defined empirically. Therefore, the alignment vector $\mathbf{a_t}$ always belong to $\mathbb{R}^{2D+1}$ (contrasting to the global attention mechanism, in which $\mathbf{a_t}$ had variable length, depending on the source sequence).

There are different possibilities to define the aligned position p_t. The most straightforward one is to assume a monotonic alignment, i. e., assume that the source and target sequences are aligned and $p_t = t$. This is, of course, not always the case, but in general, there is indeed some monotonicity between the source and target sequences.

Another possibility is the predictive alignment. In contrast to the monotonic alignment, now the model would predict the aligned position as in Equation 3.4.

$$p_t = S.\text{sigmoid}(\mathbf{v_p}^\intercal \tanh(\mathbf{W_p h_t})), \tag{3.4}$$

where $\mathbf{v_p}$ and $\mathbf{W_p}$ are model parameters to be learned and S is the source sentence length. It is possible to note that $p_t \in [0, S]$ (due to the sigmoid) and is a real number. To favor points near the predicted p_t, a Gaussian distribution centered around p_t is placed and the new alignment vector is defined by Equation 3.5.

$$\mathbf{a_t}(s) = \text{align}(\mathbf{h_t}, \bar{\mathbf{h_s}}) \exp\left(\frac{-(s - p_t)^2}{2\sigma^2}\right) \tag{3.5}$$

For instance, the standard deviation is empirically set to $\sigma = D/2$ in [20].

3.2.3 Making predictions

With the current decoder's hidden state $\mathbf{h_t}$ and its corresponding context vector $\mathbf{c_t}$ (which could be computed using global or local attention), a simple concatenation is employed, producing an attentional hidden state $\tilde{\mathbf{h_t}}$, defined by Equation 3.6.

$$\tilde{\mathbf{h_t}} = \tanh(\mathbf{W_c}[\mathbf{c_t}; \mathbf{h_t}]) \tag{3.6}$$

This attentional hidden state is then fed back to the decoder, which then produces a predictive distribution.

The idea of using attention in NMT is interesting as it overcomes the bottleneck of encoding the whole sentence in a fixed-length vector. By using attention, information flows through the whole network and the encoder is freed from the burden of encoding the whole sentence into a vector and can now produce a context disambiguated representation of the words. What is also stunning is the analysis of the alignment learned by the model. In the translation setting, the model learns some non-trivial alignments that are very characteristic of the language being considered, as it is displayed in [20, 21].

3.3 From NMT to multimodal emotion recognition

In order to adapt the ideas from NMT to the problem of multimodal emotion recognition using speech and text, the first step would be to map their intrinsic similarities and differences.

The two key concepts from NMT are the encoder-decoder architecture and the use of context vectors bridging these two units. How shall this be adapted to the multimodal setting? We use 2 networks, one for each modality. Additionally, we impose a hierarchy between them, resulting in a main and an auxiliary modality networks. The central role of the auxiliary modality network is providing the context vectors to the main modality network. The main modality network, then, makes its decision conditioned on its inputs and on the context vectors. Thus, the analogy is:

$$\text{Encoder/Decoder} \longleftrightarrow \text{Auxiliary/Main modality network}$$

The question that arises next is how should one decide on which modality is the main and which is the auxiliary? There are arguments favoring both ways. We believe that the audio could provide a better context to the text model, because the text model on its own has better results than the audio model. Moreover, one could argue that by using the textual modality as our main one, the model as a whole focuses more on the meaning of the phrase and uses the learned characteristics of the audio to help with contextualization.

To mimic the models from NMT, a first attempt could be to just initialize the hidden state of the main modality with the last hidden of the auxiliary modality.

A better approach would be to use attention mechanisms. Let the audio be the auxiliary modality and the text the main modality, each consisting of a recurrent model. From the audio RNN, we are interested in its hidden states $\bar{\mathbf{h}}_\mathbf{1}, \ldots, \bar{\mathbf{h}}_\mathbf{n}$. On the other hand, on the text RNN, at each time step we would input the vector corresponding to the current word and some mixture of the previous state and a context vector. This context vector is calculated based on the hidden states of the audio RNN. The attention weights could be computed in the same way as in the NMT, via its alignment interpretation in Equation 3.1.

Let the audio model have n RNN cells and the text model have m RNN cells. We have the hidden states of the audio model as $\bar{\mathbf{h}}_\mathbf{1}, \ldots, \bar{\mathbf{h}}_\mathbf{n}$. From them, we compute the output to the attention layer, which are the context vectors $\mathbf{c_1}, \ldots, \mathbf{c_m}$, where $\mathbf{c_t}$ is defined by Equation 3.3. The alignment

vectors $\mathbf{a_t}$ are defined according to Equation 3.1. Let $\mathbf{h_t}$ correspond to the hidden state at time step t of the text model, we then need to compute $\tilde{\mathbf{h}_\mathbf{t}}$, as in Equation 3.6.

3.4 Proposed models

The main inspiration for the developed multimodal model comes from the models that set the state of the art in the IEMOCAP dataset, presented in [16], and on the ideas exposed on Section 3.3.

The first important aspect is that all of our models are end-to-end. In contrast to [17, 16], we do not make use of hand-engineered features to learn. Instead, following the general trend in deep learning, we expect the network to learn a useful representation from the raw data to perform the classification task.

Our multimodal model is comprised of 2 sub-networks, namely the text network (considered as our main modality) and an audio network (considered our auxiliary modality). We briefly present these 2 sub-networks and then show how to combine them to assemble the proposed multimodal model.

3.4.1 Text sub-network

The text sub-network implemented is inspired in the text recurrent encoder (TRE) presented in [16].

In order to use the transcriptions from the IEMOCAP dataset, we first tokenized and indexed each transcript sample using the Natural Language Toolkit (NLTK) [22]. The tokens in a sample are, then, passed through an embedding layer, which is initialized using a pretrained embedding [23], yielding in a 300-dimensional vector per token. Then, each vector is fed to a RNN at a time step. As RNN cells, we used gated recurrent units (GRUs) [24].

Therefore, let $\mathbf{x_t} \in \mathbb{R}^{300}$ be the embedded token at time step t and $\mathbf{h_{t-1}}$ be the hidden state of the RNN at previouse time step. Then,

$$\mathbf{h_t} = f_\theta(\mathbf{h_{t-1}}, \mathbf{x_t}) \tag{3.7}$$

is the hidden state at time step t, where $f_\theta(\cdot)$ is the function representing a GRU cell parameterized by θ. Let the text RNN have m time steps. The final output of the RNN is the vector $\mathbf{o_m}$. We use this vector later, in our multimodal model to make predictions.

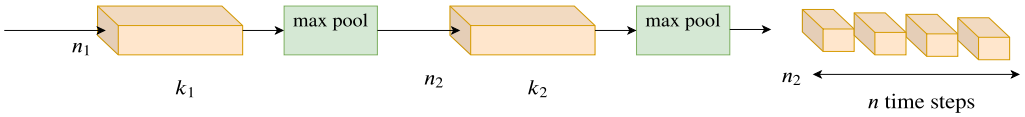


Figure 3.1: Convolutional neural network architecture for the feature extraction of the raw audio samples

3.4.2 Audio sub-network

The audio sub-network implemented is also inspired in [16], but instead of using MFCC and prosody features, we used the raw audio data, similarly to [11, 18].

The first part of the audio sub-network is responsible for feature extraction. To do so, instead of using hand-crafted filters designed to extract the features, we used learned filters in a convolutional neural network (CNN).

We start with the raw audio signal and pass it through 2 convolutional layers, which perform 1-dimensional convolutions. The first layer has n_1 filters of length k_1 and the second layers has n_2 filters of length k_2. After each convolutional layer, we perform max pooling across time, with unit stride and pooling size equals to l_1 and l_2, respectively. Figure 3.1 illustrates our convolutional layers.

As it can be observed from Figure 3.1, at the end of the CNN we have n_2 channels, each of length n, where n is defined in Equation 3.8. Then, we split this representation into n vectors of length n_2 each, as illustrated, and feed them at each time step of a RNN, as in Equation 3.7.

$$n = \left\lfloor \frac{k_2}{l_2} \right\rfloor \tag{3.8}$$

3.4.3 Multimodal model

The proposed multimodal model puts together the ideas presented in Section 3.3 with the text and audio sub-networks described. We make use of global attention, placed over all of the audio sub-network's hidden states. From them, we build a context vector that is used at each time step by the RNN cells of the text sub-network.

Given a text sample and its corresponding audio, we first feed the audio through the audio sub-network. By doing so, we have produced the hidden states for each time step of the audio sub-network $(\bar{\mathbf{h}_1}, \ldots, \bar{\mathbf{h}_n})$, considering that the audio RNN has n time steps.

Then, we initialize the hidden state for the text sub-network with the

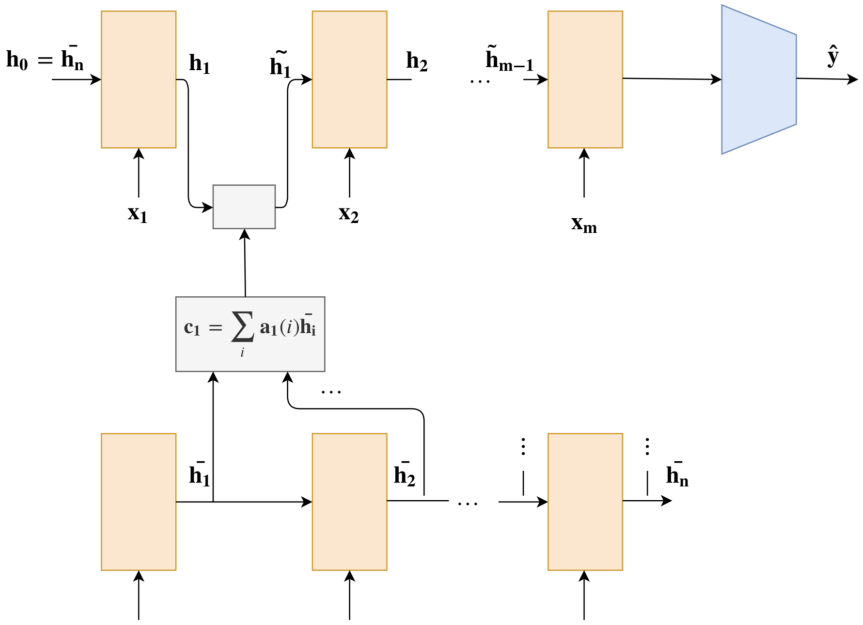


Figure 3.2: Proposed multimodal model with attention. On the upper part, is the text RNN, in the lower, the audio RNN and the attention layers bridgind both.

last hidden state from the audio sub-network $\bar{\mathbf{h}}_\mathbf{n}$ and feed the first embedded token $\mathbf{x_1}$, producing hidden state for the text sub-network, according to Equation 3.7.

From there on, we calculate the score between the current text hidden state and all of the audio's RNN hidden states. We explored different score functions and ended up choosing the same metric as [21], i. e., a feedforward neural network, shown in Equation 3.2. The vector $\mathbf{v_a}$ and the matrices $\mathbf{W_{a1}}$, $\mathbf{W_{a2}}$ are learned.

Then, we use Equation 3.1 to compute the attention weights and produce the context vector according to Equation 3.3. With the context vector and the text hidden state, we produce $\tilde{\mathbf{h}}_\mathbf{t}$ as in Equation 3.6. The transformed hidden state $\tilde{\mathbf{h}}_\mathbf{t}$ is then fed to the next text RNN cell.

At the end, the text RNN produces the output vector $\mathbf{o_m}$. This vector is, then, passed through a fully connected layer with a softmax activation producing the vector $\hat{\mathbf{y}}$ with the class probabilities and the label is predicted based on the most likely class. Figure 3.2 depicts our multimodal model.

The loss function we minimize is given by Equation 3.9, where C is the total number of classes, N the number of training samples, $\mathbf{y_i}$ is the true

label vector and $\hat{\mathbf{y}}_\mathbf{i}$ the class probabilities from the softmax.

$$\mathcal{L} = -\log \prod_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log \left(\hat{y}_{i,c}\right) \tag{3.9}$$

Chapter 4

Experiments and Emprirical Results

4.1 Training details

We train and assess our model in the IEMOCAP dataset, making use of the audio and the trasncriptions available. To be consistent with previous research, we followed the procedures described in Chapter 2. We split the dataset into training, testing and validation sets, in a proportion of 8:0.5:1.5, respectively. The hyperparameters are selected based on extensive search in the validation set.

4.1.1 Text model

We train a text-only model using the architecture described in Section 3.4.1 and feeding the output of the RNN to a fully connected layer with softmax activation, to predict the class probabilities. The text model evaluated is essentially the same as the one presented in [16].

We use gated recurrent units (GRUs) as RNN cells and our text RNN has only one layer with 200 neurons. The maximum number of time steps of the RNN is 128, which is the length of the longest trasncript sequence. We inialize the weights of the hidden units orthogonally and the text embedding layer is initialized with pretrained GloVE word embeddings [23].

We use a batch size of 32 and zero-pad all of the samples to have the same length. We use the Adam optimizer with a learning rate of 3.10^{-3}. In the RNN, we use dropout with dropout probability equal to 0.55. We train the text model for 10 epochs.

4.1.2 Audio model

Similarly to the text-only model, we train an audio-only model with the architecture described in Section 3.4.2 followed by a fully connected layer with softmax activation. The RNN part of the audio-only model is inspired on [16], but we do not make use of hand-crafted audio features.

Our CNN has two layers. We use 25 and 50 filters with lengths 25 and 5, respectively. After each convolutional layer, we apply max pooling across time with pool size of 50.

Similarly to the text-only model, we use GRU cells in the RNN with one layer with 200 neurons. We use a batch size of 32 and zero-pad all of the samples to have the same length. We use the Adam optimizer with a learning rate of 3.10^{-3}. In the RNN, we use dropout with dropout probability equal to 0.55. The audio model is trained for 50 epochs, which is higher than the text model due to the increase number of parameters introduced.

4.1.3 Multimodal model

For the multimodal model, we put together the text-only and the audio-only networks and perform the classification based on the output of the text RNN, which is fed to a fully connected layer with softmax activation, as described in Section 3.4.3.

We significantly reduce the networks' complexity, especially the audio network. We still use two layers in the CNN, but with 4 and 8 filters of lengths 25 and 5, respectively. The text and audio RNNs have a single layer, both with 200 neurons. Also, we increase the dropout in the audio RNN to 0.8, while keeping the dropout in the text RNN equal to 0.55. We used a batch size of 32 and a learning rate of 3.10^{-4}. We trained the model for 1000 epochs.

To calculate the attention score, we used Bahdanau's attention [21]. We also tried to use Luong's attention [20], but it yielded in worst results. We have hypothesized as to why this was the case and we discuss it in the Analysis section.

All of the models were implemented on Python using Tensorflow. The code is openly available, along with instructions on how to run it [1].

[1] `https://github.com/gustavocidornelas/fused-multimodal-emotion`

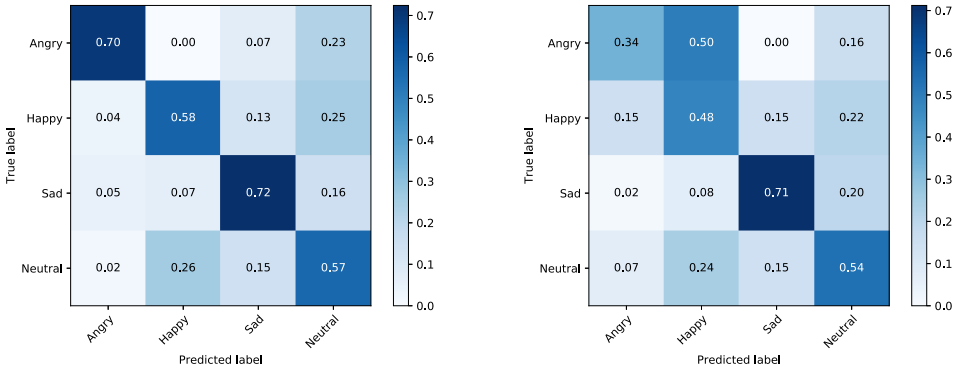
Table 4.1: Model performance comparisons.

	Mean accuracy
Text model (implemented from [16])	**0.625**
Text recurrent encoder [16]	0.635
Audio model (ours)	**0.527**
Audio recurrent encoder (ARE) [16]	0.546
ACNN [8]	0.561
Triparthi *et al.* [17]	0.557
Multimodal model (ours)	**0.610**
Multimodal dual recurrent encoder (MDRE) [16]	0.718
Multimodal dual recurrent encoder with attention (MDREA) [16]	0.690

4.2 Performance evaluation

Table 4.1 summarizes the results obtained with each of the investigated models.

Besides the accuracies shown in Table 4.1, we can observe for each of the implemented models the confusion matrix, in Figures 4.1 and 4.2.



(a) Text model.

(b) Audio model.

Figure 4.1: Confusion matrices for test set for the text and audio models.

4.3 Analysis

First, we start with the analysis of the accuracies presented in Table 4.1. The text model shows a better performance than the audio model, which demonstrates that the textual data is informative in the task of emotion recognition. As it can be noted, our audio model performed on the same

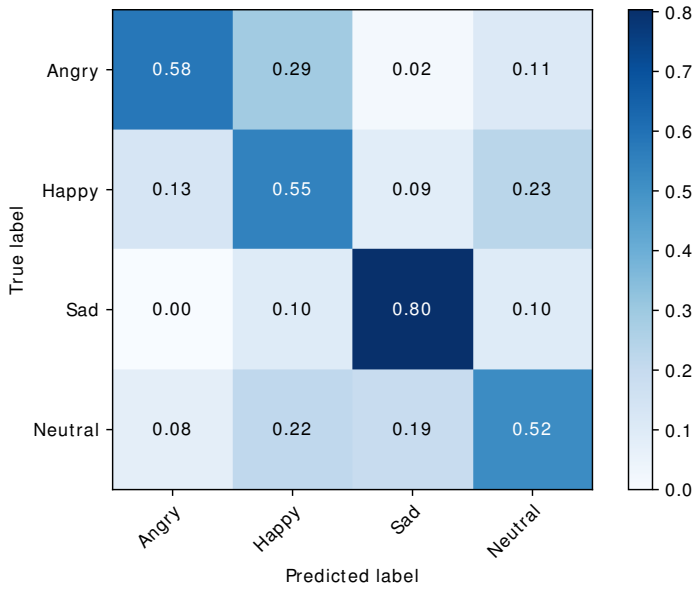


Figure 4.2: Confusion matrix for the teste set for the multimodal model.

level as the models that set the state-of-the-art for the IEMOCAP dataset. This result reinforces our point that the convolutional layers placed at the beginning of our audio model learns to extract informative features for the task, without relying on engineered features, such as the MFCC or prosody.

A significant limitation, especially for our audio model and multimodal models, is the size of the IEMOCAP dataset. We believe that increasing the size of the convolutional layers can yield in a significant improvement in our models' performance and probably surpass that of the models that make use of features. This could only be done with a larger dataset, as increasing the convolutional layers with the IEMOCAP dataset quickly leads to overfitting.

In our multimodal model, we investigated different possibilities for the calculation of the attention score. Following [20], we started with the inner product, as in the top part of Equation 3.2. This choice yielded in poor results in terms of classification accuracy. The idea of defining a score between the hidden states of the two modalities is to measure their similarity and learn an alignment. Doing this through the inner product makes sense for a problem such as NMT, because the hidden states are generated from data that comes from the same embedding, in that case, text. The problem is that in the multimodal setting, we wish to combine hidden states that

were generated from very different embeddings: one comes from textual and the other from audio data. The poor results using the inner product suggests that, *a priori*, there is no relation between the direction of the hidden state vectors and their content. Therefore, we decided to calculate the score as in [21], shown in the bottom part of Equation 3.2, using a feedforward neural network. We base our choice in the assumption that a neural network would be capable of learning the alignment function that we look for. This yielded in better results, as demonstrated by our final multimodal model in Table 4.1.

Our multimodal model's performance is lower than that of the multimodal models in [16]. Also, it is notable that our multimodal model performed better than our audio model, but worse than the text model. What is possible to observe during the training of our multimodal model is that the model quickly overfits the training data. We made an effort to decrease the model's complexity and increase regularization using dropout in the RNNs, but overfitting is still a problem. The IEMOCAP dataset, although considered one of the most complete multimodal datasets for emotion recognition, is still of limited size and we believe our model is too complex for the data available.

From the confusion matrices, we can note that for all of the models, most of the classes most frequently make mistakes with the *neutral* class. This is known as the *neutral* class misclassification bias. Although not accentuated in our models, this bias exists and the reason is discussed in [8], where it is shown that the *neutral* class is located in the center of the activation-valence space, which makes the difference between *neutral* and the other classes harder to identify. From the confusion matrix of the audio model, specifically, it is possible to observe the big confusion between the *angry* and *happy* classes. This probably happens because although *angry* and *happy* are on two opposite extremes, their audio waveform is not that different from one another. This big confusion does not exist in the text model and is reduced in the multimodal model.

Our multimodal model differs significantly from the multimodal models in [16]. First, the multimodal model without attention in [16] is comprised of two separate RNNs with GRU cells, one for the audio and one for the text. These two RNNs are merged only at their end via a fully connected layer. Second, in the model with attention, the attention is only computed between the output from the audio RNN and the hidden states of the text RNN. After that, the generated context vector is merged with the output from the audio RNN via a fully connected layer. This usage of attention is, thus, conceptually different than the one deployed in our model. In terms of the RNN's hyperparameters, we started the tuning pro-

cedure with the same hyperparameters as [16], but then differed in order to maximize the accuracy in the validation set, until we settled with the parameters described in Sections 4.1.1, 4.1.2 and 4.1.3.

Chapter 5

Conclusion

In this work, we explored the problem of emotion recognition from a multimodal perspective, using audio and text data. We proposed a architecture inspired in the encoder-decoder structure with attention that is deployed in NMT. The proposed model is fully end-to-end and does not make use of hand-engineered features. We combine the different modalities networks in a novel way, aiming to explore to the fullest the the most relevant aspects of each modality and their interactions. Our multimodal model's performance is bellow the state-of-the-art and we believe that this is due to the small size of the IEMOCAP dataset and our model's complexity.

There is definitely room for improvement and future work can explore different alternatives with the proposed model. One could try implementing local attention, instead of global attention, reducing, thus, the number of parameters to be learned. Also, future work can explore the effect of switching the roles of main and auxiliary modality networks. Additionally, one could investigate an architecture where both networks are recurrently fused, without imposing a hierarchy between them. Another interesting line of research would be investigating how to best quantify the similarity between the hidden state vectors of both modalities, taking into account the fact that these vectors were constructed from different data embeddings.

Bibliography

[1] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: A review," *Int. J. Speech Technol.*, vol. 15, no. 2, pp. 99–117, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1007/s10772-011-9125-1

[2] B. Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in *Proceedings of the 2003 International Conference on Multimedia and Expo - Volume 2*, ser. ICME '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 401–404. [Online]. Available: http://dl.acm.org/citation.cfm?id=1170745.1171501

[3] T. Seehapoch and S. Wongthanavasu, "Speech emotion recognition using support vector machines," in *2013 5th International Conference on Knowledge and Smart Technology (KST)*, Jan 2013, pp. 86–91.

[4] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," 03 2017.

[5] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong, "The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, April 2016.

[6] Y. Wang, L. Neves, and F. Metze, "Audio-based multimedia event detection using deep recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 2742–2746.

[7] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition." ISCA - International Speech Communication Association, September 2015.

[8] M. Neumann and N. T. Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," *CoRR*, vol. abs/1706.00612, 2017. [Online]. Available: http://arxiv.org/abs/1706.00612

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751. [Online]. Available: https://www.aclweb.org/anthology/D14-1181

[10] J. N. Bassili, "Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face," *Journal of personality and social psychology*, vol. 37, pp. 2049–58, 12 1979.

[11] P. Tzirakis, G. Trigeorgis, M. A. Nicolaou, B. W. Schuller, and S. Zafeiriou, "End-to-end multimodal emotion recognition using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1301–1309, Dec 2017.

[12] S. Tripathi, S. Acharya, R. D. Sharma, S. Mittal, and S. Bhattacharya, "Using deep and convolutional neural networks for accurate emotion classification on deap dataset," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, pp. 4746–4752. [Online]. Available: http://dl.acm.org/citation.cfm?id=3297863.3297883

[13] T. Baltrusaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *CoRR*, vol. abs/1705.09406, 2017. [Online]. Available: http://arxiv.org/abs/1705.09406

[14] F. Lingenfelser, J. Wagner, and E. André, "A systematic discussion of fusion techniques for multi-modal affect recognition tasks," in *Proceedings of the 13th International Conference on Multimodal Interfaces*, ser. ICMI '11. New York, NY, USA: ACM, 2011, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/2070481.2070487

[15] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L. Morency, "Tensor fusion network for multimodal sentiment analysis," *CoRR*, vol. abs/1707.07250, 2017. [Online]. Available: http://arxiv.org/abs/1707.07250

[16] S. Yoon, S. Byun, and K. Jung, "Multimodal speech emotion recognition using audio and text," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2018, pp. 112–118.

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

[18] G. Lima and J. Bak, "Speech emotion classification using raw audio input and transcriptions," in *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*, ser. SPML '18. New York, NY, USA: ACM, 2018, pp. 41–46. [Online]. Available: http://doi.acm.org/10.1145/3297067.3297089

[19] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, no. 4, p. 335, Nov 2008. [Online]. Available: https://doi.org/10.1007/s10579-008-9076-6

[20] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015. [Online]. Available: http://arxiv.org/abs/1508.04025

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv e-prints*, vol. abs/1409.0473, Sep. 2014. [Online]. Available: https://arxiv.org/abs/1409.0473

[22] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[23] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *In EMNLP*, 2014.

[24] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555