Small and large uni and multi modal language models for sentiment analysis: a comparison
NLP Course Project

Davide Femia, Riccardo Paolini, Alessandro D'Amico and **Sfarzo El Husseini**
Master's Degree in Artificial Intelligence, University of Bologna
{ davide.femia, riccardo.paolini5, alessandro.damico5, sfarzo.elhusseini }@studio.unibo.it

Abstract

The purpose of this paper is to provide guidelines for implementing a multimodal model that includes textual and audio features. Specifically, our focus is on the differences between small and large language models: we compare them in terms of performances for a Sentiment Analysis task (Emotion Recognition on the IEMOCAP dataset). In order to highlight the advantages and disadvantages of each approach and to give a meaningful evidence of the differences between the two types of models, we implement and compare the scores among the single modality models (audio or text) and a bimodal model that integrates the best one for each modality, finally analyzing the effectiveness of classic fusion methods.

1 Introduction

Sentiment analysis is the study of extracting emotions and opinions from text data and its recent popularity is due to its applications such as monitoring social media, managing brand reputation, and improving customer service. The objective of this popular task can be to identify emotions(happiness, anger, ...) or categorize text (positive, negative or neutral).
Bimodal sentiment analysis instead has the same objective but uses both text and audio to better understand emotions. By combining the two input modalities, models can capture both wording and tone of voice to improve accuracy, the key challenge is effectively merging this features to create robust and precise representations for predicting emotions.
Bimodal models typically have separate parts for processing each modality before combining them through fusion techniques. Fusion is classified into:

- early: mixes features from each modality before inputting them into a shared classifier.
- late: train separate classifiers for each modality and combines their outputs.
- hybrid: combines early and late fusion, often with deep learning to learn complex hierarchical representations.

In the literature it is possible to find several methods for sentiment analysis, the most popular approaches based on text use different ways to tackle the problem of word embedding applied to tokens. Word embedding is the process of converting words into vector representations of a point in space with a specific dimension; there are three main ways to produce these embeddings:

- Word2Vec/Glove: based on bag-of-words / skip-grams (Word2Vec). Based on word-word co-occurrence matrix (Glove).
- Neural representations: based on neural networks.

In general, embeddings obtained using neural networks are richer than the ones obtained via Word2Vec or Glove, as they are able to learn very complex representations, but at the same time they are slower and more computationally expensive, as they have to be learned. Some of the best performing models for emotion detection from text are Bidirectional Long Short- Term Memory (BiLSTM) and Transformers like BERT. In particular, the first model can use Glove or Word2Vec representation as initial weights of an embedding layer (it is also possible to freeze the weights of the embedding), while Transformers have their own internal embedding.
With regard to audios instead, the choice of the features to be extracted is directly related to the model intended to use. Below we highlight the main kinds of features that can be extracted from audios:

- Statistical features: using only audio statistics such as energy, spectral centroid, spectral flux, zero-crossing rate, amplitude envelope.
- Frequency features: representing the signal in the frequency domain, mainly using Mel-frequency cepstral coefficients (MFCC) or Mel-Spectrogram (MEL).

- Deep features: extracting a latent representation via a Convolutional Neural Network.

The most popular approaches based on audios mainly uses frequency features and Deep features and are respectively CNN models and audio transformers.
In our work we decided to implement several early fusion models which differs from the way the representations are obtained and how the representations are combined. To understand the effectiveness of the representations produced by single-mode architectures, we decided to train them for the sentiment analysis task by applying a classification head. The single modality architecture tested in our work are 6 (2 for text, and 4 for audio). After evaluating each of the single-mode architectures, we use the best on text and the best on audio by combining the extracted representations in three different ways. These bimodal architectures are then tested with different fusion modes to see which one is the best. Finally, we highlight the errors made by the model and how it can be improved. All the models were evaluated on IEMOCAP. In our experiments, bimodal models, thanks to their ability to combine different forms of data, proved to be better than unimodal models.

2 Background

In order to better describe our work, we provide a brief overview on text models and audio models.

2.1 Text models

We will see two type of text models, BiLSTM and text transformers, we will focus our description on the BERT transformer.

2.1.1 BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) is a type of recurrent neural network that allows the input to flow in both directions thanks to the introduction of an additional LSTM layer where the input sequence is processed in reverse. This allows it to leverage the information from both the past and future contexts, making it highly effective for capturing sequential dependencies between words and phrases. The outputs from both networks are then combined using operations such as averaging, summing, multiplying, or concatenating.

2.1.2 BERT

The Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. is essentially a Transformer Encoder which is composed by a stack of Encoder layers that capture word relationships.
Each encoder layer is composed by:

1. MultiHeadAttention: essentially computes multiple SelfAttentionMechanisms (the number of SelfAttentionMechanisms is called num_heads) and combines them with a concatenation followed by a Linear layer that combines the output of each head.
2. Add and Normalize: sum the output of the MultiHeadAttention with the value input of the MultiHeadAttention providing some skip connections and then normalize the result.
3. Feed Forward: consists of two Dense layers with a ReLU activation in-between, and a dropout layer.
4. Add and Normalize: another step similar to 2 the input bringed forward this time is the input to the Feed Forward block.

The SelfAttentionMechanism is a process that it's like a fuzzy, differentiable, vectorized dictionary lookup. Essentially given a query, a key and a value input produces three vectores with three different set of weights (one for each of the input) than combine the vectors produced by computing a scaled dot product between the vectors produced from query and key inputs. The computed scaled dot product is used to get some attention scores by applying a softmax, than the attention scores are used in combination with the produced value vectors in order to get the output of the entire SelfAttentionMechanism.
The embeddings in input to the entire Transformer Encoder is the output of an Embedding Layer combined with a Positional Embedding in order to enrich the embeddings with positional information (in practice if the attention scores are computed only on positional embedding we get that each position is similar to neighboring positions).
The advantages of using the "Attention" mechanisms are the creation of a very rich contextual embedding and the excellent ability to find long-term dependencies. To use BERT as a classifier, a classification head must be placed over the embeddings produced.

2.2 Audio models

As we mentioned previously the model we use is linked to the features we extracted. For instance, if we work with frequency features a Convolutional

Neural Networks (CNNs) or networks based on Long Short-Term Memory cells (LSTM) are more suitable given that a lot of information is contained in the temporal evolution of those. Transformers, instead, are the models to be used with deep features.

2.2.1 Convolutional Neural Networks

A convolutional neural network is a type of neural network organized hierarchically, composed of multiple sequential layers. In a typical model, there are several convolutional layers that process visual information. These layers apply a set of learned filters to the input data, resulting in a collection of feature maps.
Badshah et al. proposed a method for emotion classification based on a CNN with spectrograms as input. Specifically, they extract several spectrograms for each audio signal, then classify each of them independently and finally combine these predictions to generate a single final prediction.

2.2.2 Wav2Vec 2.0

Wav2Vec 2.0 proposed in (Baevski et al., 2020) is a model for self-supervised learning of representations from raw audio. It is composed of a multi-layer convolutional feature encoder $f : X \rightarrow Z$ that given a input audio produces $z_1, \cdots, z_T$ latent speech representation, where T is the number of time-steps. This sequence of latent vectors is then passed to a context network composed by a Transformer $g : Z \rightarrow C$ which creates representations $c_1, \cdots, c_T$ capturing information from the entire sequence. The latent vector z is also processed by a quantization module which discretize it to a set of speech representations via product quantization, i.e. method approximating nearest neighbor search. The model is than trained using contrastive learning which aims at learning low-dimensional representations using unlabeled data.

3 System description

In our work we implemented a multimodal conversational model, the two inputs of our models are essentially audio and textual transcription of some utterances. Essentially the model we implemented is capable of choosing the size of the conversation as an hyper-parameter, in order to do so we expect as input to our model a set of texts and audios, each set is stacked along a new dimension, the last element of this new dimension is to be considered as the current audio/text the other elements has to be considered as past history. The model is capable to take into account each audio/text separately by means of an audio/text embedder in order to get an audio/text embedding. the audio and text embedders can be implemented in different ways, we decided to use the following embedders:

- Audio Embedder
  - Simple: this embedder expects as input some frequency features extracted from audios (MFCC, MEL and Chroma) and apply a set of 1D convolutions in order to extract some high-level features from the input features, then apply a stack of BiLSTM in order to get a contextual representation looking at previous and successive high-level features. The last layer of the stack extracts an overall embedding of the entire audio.
  - Wav2Vec 2.0: described in previous section, it is used to get a rich contextual embedding of audio sub-parts, this embeddings are combined with a pooling strategies in order to get an overall embedding of the entire audio.
- Text Embedder
  - Simple: this embedder expects as input some tokens extracted by a custom tokenizer, it is composed by an embedding layer followed by a stack of BiLSTM in order to get a contextual representation looking at previous and successive token embeddings. The last kayer of the stack extracts an overall embedding of the entire sentence.
  - ALBERT: essentially is a lighter version of BERT that is used to get a rich contextual embedding for each token, this embeddings are combined with pooling strategies in order to get an overall embedding of the entire sentence.

Regardless on how these embedders are implemented the audio and text embeddings produced will be combined with different fusion methods (Concatenate, Add, Cross-Attention) in order to get a fusion embedding for each turn of the conversation.
The fusion embeddings will be fed by a forward LSTM layer in order to get the context from past

turns, this context will be combined with the fusion embedding of the current turn by means of a residual skip connection that helps in preventing the vanishing/exploding gradient problem.
The output of the residual skip connection is then fed to a Dense layer with ReLu activation followed by a Dense layer with activation softmax.
The entire architecture with the best parameter configuration that maximize the $F1_{weighted}$ metric is shown at Figure 1.

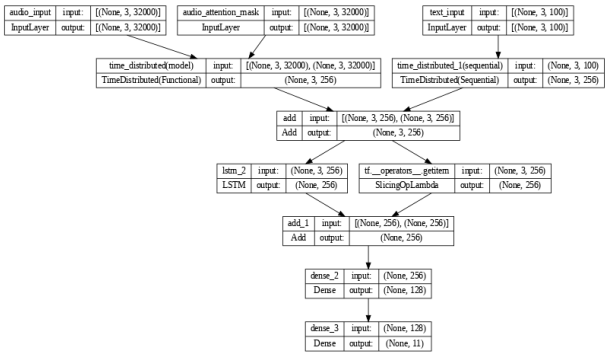

Figure 1: best model architecture BiLSTM+Wav2Vec2.0 with 2 seconds audio inputs, Add fusion, $conv_{length} = 3$ trained with *classweights*. The *TimeDistributed* Layers contain the audio and text embedders. The *SlicingOpLambda* simply extract the fusion embedding of the current utterance

4 Data

To address the described comparison task, the IEMOCAP dataset was chosen (Busso et al., 2008). IEMOCAP is a widely used and well-known collection of multimodal data, valuable for investigating the correlation between speech, gestures, and facial expressions in conveying emotions. Therefore, it is a good tool to test the performance of the interested modalities (text and audio) and to test the effectiveness of their combination in the architecture of the models. The recorded dialogs contained in IEMOCAP are either improvisations of affective scenarios, or performances of theatrical scripts that have been manually segmented into utterances. Each utterance from either of the actors in the interaction has been evaluated categorically over the set of: *angry, happy, sad, neutral, frustrated, excited, fearful, surprised, disgusted, other* by at least three different annotators. This dataset also offers a VAD (Valence, Activation, Dominance) labeling, that however is not orthogonal to the listed emotion, but is rather a subsitute of them, therefore we didn't make use of this.

The dataset was split by sessions (as described in Table 1) and just the audio and the textual transcription were used, while the video and the motion-capture (MoCap) were not employed. Since the

Session	Set	N.utterances	N.dialogs
1	Train	1085	28
2	Train	1023	30
3	Train	1151	32
4	Validation	1031	30
5	Test	1241	31

Table 1: IEMOCAP split used in this work

dataset is highly unbalanced (Figure 2), just 6 out of 11 emotions were used for benchmarking, as done in other works (Joshi et al., 2022)(Dutta and Ganapathy, 2023). Audios were used in Wav2Vec

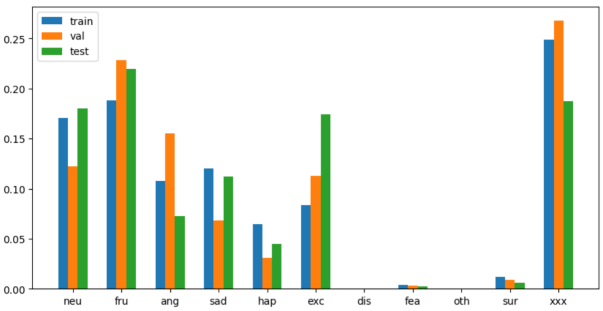

Figure 2: IEMOCAP, distribution of each emotion inside their split (all emotions of the same split sum up to 1). This visual includes also the minority classes (*dis*, *fea*, *sur*, *oth*) and the ambiguous classifications (that are those ones on which there isn't a majority agreement for the annotators, marked as *xxx*), which have been masked.

testing different limits for the duration (respectively of 1,2,3 and 4 seconds), because of the increasing need of memory when preprocessing the whole dataset at once. As preprocessing step, we've implemented the concept of *conversation* considering the sequence comprising the current utterance and the preceding $n-1$ ones, both for the text and the audio input (that is what we mean for $conv_{length} = n$).[1] The sequences composed by the previous n-1 utterances and the actual one are fed to the model by stacking them on a new dimension as described in the previous section.

[1] If inside the current dialog the utterance has less than $n-1$ preceding utterances, the missing ones are substituted simply by padding.

5 Experimental setup and results

The models were trained in the standard approach using the *CategoricalCrossentropy* as a loss considering all the relevant classes (Section 4). We've also implemented and tried a variant of the loss computation that uses *classweighting*, with weights w_i computed according to the following formula:

$$w_i = \frac{N}{|C| \cdot N_i}$$

Where N refers to the size of the dataset including only the relevant classes, $|C|$ refers to the number of relevant classes considered, N_i refers to the support of the i^{th} class in the dataset, this heuristic is inspired by (King and Zeng, 2001). In this way we give more weight in the loss for the minority classes and less weight for the majority classes: it is an attempt of driving the model to focus more on the performance on the minority classes rather than the majority ones, trying to improve final results.
To score the results for each architecture tested, $F1_{weighted}$ was mainly used[2].
All the models were run using *Adam* as Optimizer (with $learning_{rate} = 0.001$), and a machine having 25 GB of RAM and 16 GB of VRAM (Nvidia T4). Because of higher computational resources requirements, multimodal models were trained and tested on a much powerful machine, having an Nvidia A100, in that case we've used $batch_{size} = 32$. As early stopping procedure we've monitored the Validation's $F1_{weighted}$ and set a *Reduce Learning Rate On Plateau* procedure with $factor = 0.4$. The F1-average scores are reported in the Tables 2, 3, 4 and in Figures 3, 4, 5 for an easier visualization, which also sums up the dependency on $conv_{length}$ of the implemented models. All the results shown are averaged with two runs on different seeds (42, 77) to ensure reproducibility. We didn't choose any model as a baseline because we wanted to compare our models in different settings (audio unimodal, text unimodal, bimodal).

6 Discussion

To compare results between the various models (Figure 3), we've selected the ones built with a $conv_{length} = 3$, this is not the conversation length for which we can obtain the best results, however, the computational resources at our disposal are limited and we were interested to do this comparison

[2] $F1_{weighted}$ is computed as $F1_{weighted} = \sum_{i \in C} \frac{N}{N_i} \cdot F1_i$.

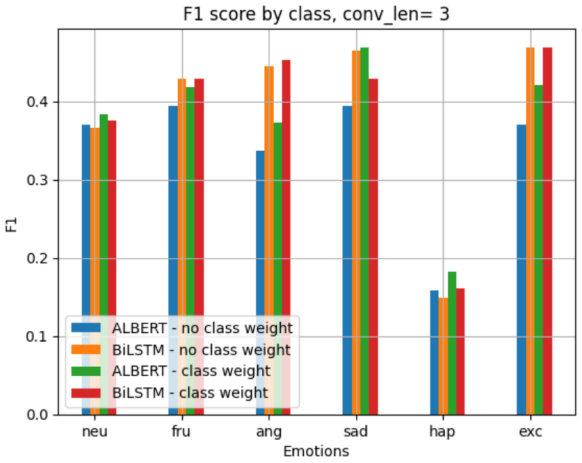

Figure 3: $F1$ metric measured on Test set for the different emotion classes for the implemented text models, with $conv_{length} = 3$

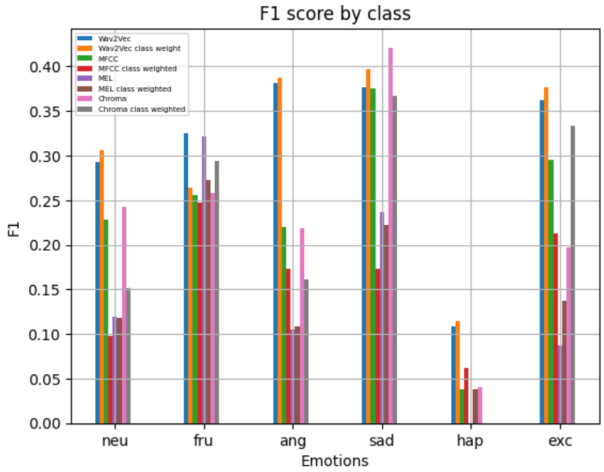

Figure 4: $F1$ metric measured on Test set for the different emotion classes for the implemented audio models, with $conv_{length} = 3$

with models that we could actually pair.
ALBERT and the simple BiLSTM model are quite comparable: The BiLSTM model performs better than ALBERT in most of the categories: a reason for this could be found in the fact that the two models haven't been previously trained to have emotional content encoded in their weights and while BiLSTM is learning this setting starting from a random state, ALBERT does it starting from its pretrained status, that is possibly encoding different insights. The effect of the usage of class weighting on the BiLSTM model is the improvement of the performance on the minority classes in the train set (respectively: *angry*,*sad* and *happy*), while gives a general improvement in the F1 scores of ALBERT. The scores increase with the increasing length of

	Emotion					
Model	**neu**	**fru**	**ang**	**sad**	**hap**	**exc**
ALBERT	0.37	0.39	0.34	0.39	0.16	0.37
ALBERT (cw)	0.38	0.42	0.37	0.47	0.18	0.42
BiLSTM	0.37	0.43	0.44	0.46	0.15	0.47
BiLSTM (cw)	0.38	0.43	0.45	0.43	0.16	0.47

Table 2: $F1$ metric measured on Test set for the different emotion classes for the implemented text models, with $conv_{length} = 3$

.

	Emotions					
Model	**neu**	**fru**	**ang**	**sad**	**hap**	**exc**
MFCC	0.23	0.26	0.22	0.37	0.04	0.29
MFCC (cw)	0.10	0.25	0.17	0.17	0.06	0.21
MEL	0.12	0.32	0.10	0.24	0.	0.09
MEL (cw)	0.12	0.27	0.11	0.22	0.04	0.14
Chroma	0.24	0.26	0.22	0.42	0.04	0.20
Chroma (cw)	0.15	0.29	0.16	0.37	0.	0.33
Wav2Vec	0.29	0.32	0.38	0.38	0.11	0.36
Wav2Vec (cw)	0.31	0.33	0.38	0.40	0.11	0.37

Table 3: $F1$ metric measured on Test set for the different emotion classes for the implemented audio models, with $conv_{length} = 3$

the conversation history, this is because conversations usually have a dominant emotion along their temporal dimension and because in *IEMOCAP* conversations include just two speakers. An alternative to this approach could have been considering also the future utterances, as done in (Kim and Vossen, 2021) (with respect to the one actually asked to classify), but we preferred to consider this problem as a classification task that has to be done just considering past sentences (apart from this aspect, this is not possible with the current architecture chosen for the models).

Taking in consideration the performance of the different audio models (Figure 4), With respect to the text models, audio models have generally lower scores and seems extracting insights which are less effective. The effect of class weighting is similar to the one seen for text models: for the simple lightweight models can improve performance on

	Emotion					
Fusion method	**neu**	**fru**	**ang**	**sad**	**hap**	**exc**
Add	0.42	0.41	0.49	0.48	0.18	0.52
Add (cw)	0.42	0.43	0.49	0.50	0.21	0.52
Concat	0.39	0.40	0.48	0.53	0.21	0.50
Concat (cw)	0.39	0.42	0.47	0.47	0.20	0.55
Cross Att	0.27	0.35	0.45	0.50	0.18	0.47
Cross Att (cw)	0.30	0.41	0.47	0.45	0.17	0.43

Table 4: $F1$ metric measured on Test set for the different emotion classes for the implemented bi-modal models *BiLSTM (text) + Wav2Vec (audio)*, with $conv_{length} =$ 3Bi-modal model [$conv_{length} = 3$

.

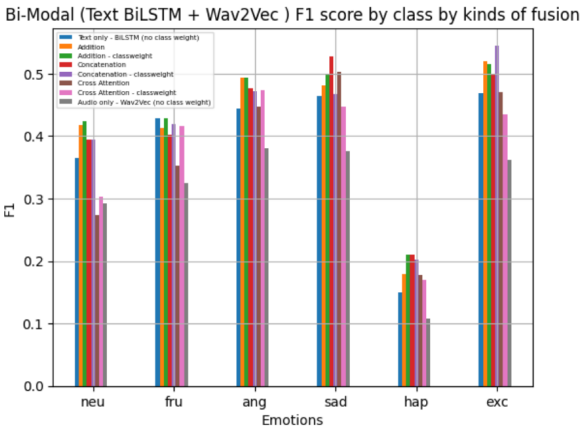

Figure 5: $F1$ metric measured on Test set for the different emotion classes for the implemented bimodal (audio+text) models, with $conv_{length} = 3$

certain minority classes but can also worsen performance on majority classes, while for *Wav2Vec* gives a general improvement. The *happy* class seems, even in this case, being the worst predicted class. The only case in which *Chroma* overperforms *Wav2Vec* is for the *sad* emotion: Chroma seems giving comparable results even having a much more simple structure.

We performed train/test loops using different cuts for the audio track fraction used for Wav2Vec (Figure 9) [3] and, as expectable, the more we consider of each audio, the better the performance. Of course,

[3]Since we can't use the whole length of the audios for computational resources limits, we've experimented 1,2,3 and 4 sec (0.005, 0.1, 0.25 and 0.5 quantiles circa).

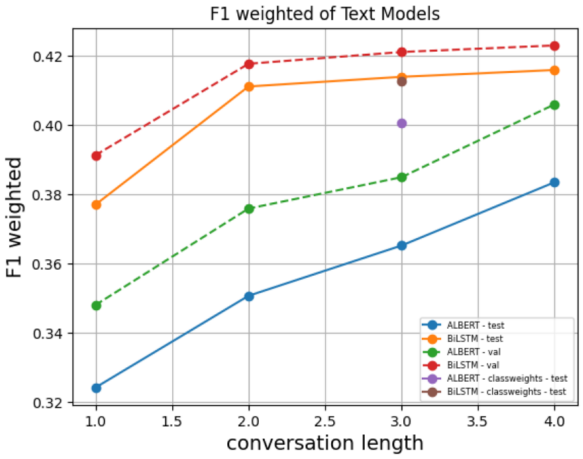

Figure 6: $F1_{weighted}$ cumulative metric measured on Test set for the implemented text models using $conv_{length} = \{1, 2, 3, 4\}$

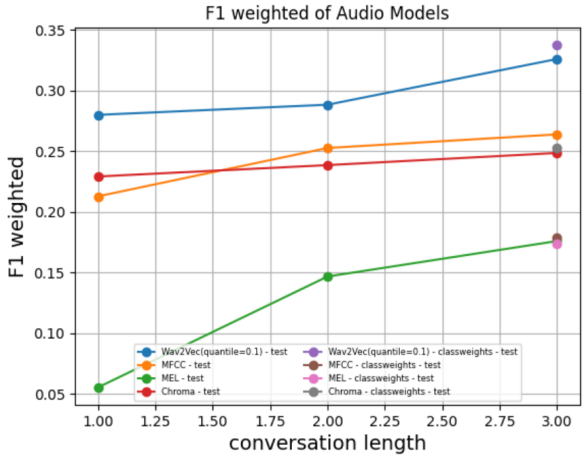

Figure 7: $F1_{weighted}$ cumulative metric measured on Test set for the implemented audio models using $conv_{length} = \{1, 2, 3, 4\}$

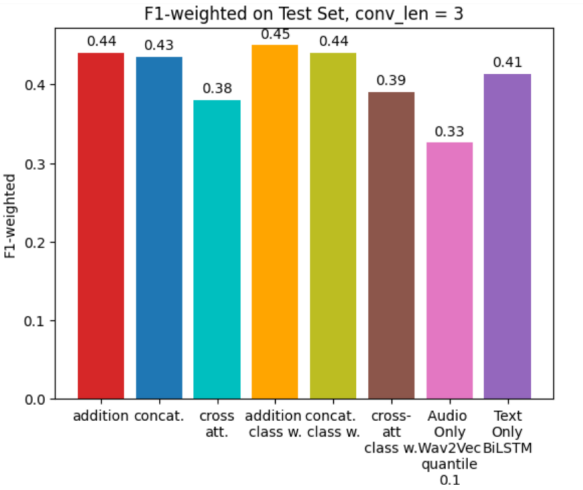

Figure 8: $F1$ metric measured on Test set for the different emotion classes for the implemented bimodal (audio+text) models, with $conv_{length} = 3$

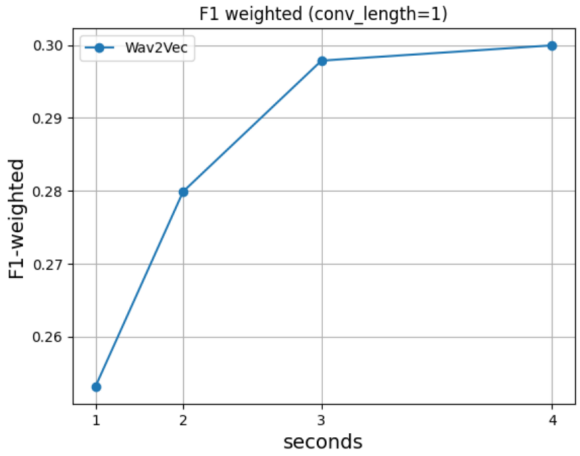

Figure 9: $F1_{weighted}$ metric measured on Test set for Wav2Vec using the first 1,2,3 and 4 seconds of each audio

one could randomically choose the starting time of the fragment of the track, but to simplify our experiments we've decided to take the fragment starting from the actual beginning of each track. For the bi-modal model, *Addition* and *Concatenation* resulted being the methods giving the best performances, combining the advantages of both of the models. *Cross-Attention* has instead resulted in being the least effective fusion method, probably leading to a sort of "confusion" that causes the model to have lower performances (similar to using *Wav2Vec* only).
The bimodal model has been able to give scores higher than the ones obtained with the single modalities (text only or audio only), however, even if better scores were obtained, the model doesn't seem completely able to capture the aspects that should be captured just from the mixed modalities.
One of the wrong classifications given by our best model (Figure 1) is:

- input text: *[laughter] oh, i'm going to be working too, i mean.*
- audio input: waveplot is shown in Figure 10
- true label: *exc*
- predicted label: *hap*

The source of this error could be the fact that the excited sentiment could be difficult to distinguish from the happy sentiment, also the length of the audio is only 2 seconds therefore the audio input only hears at the first laugh of the actor and some

other words but not the entire audio. the text instead is processed in its entirety, this could lead into very different representations and lead to confusion.

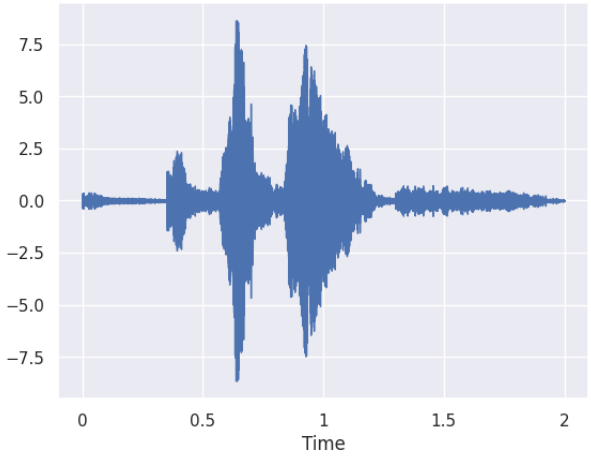

Figure 10: audio input waveplot of a wrongly classified sample

7 Conclusion

Audio models have resulted being not good enough to provide useful insights that could be aggregated to the text modality in the sentiment analysis task. Another limitation has been that the implemented fusion mechanisms were too simplistic: this problem could be in part overcome using custom fusion architecture as the one proposed by (Majumder et al., 2018). Moreover, the required resources necessary to experiment the proposed modalities were high, but at the same time not sufficient to run all the possible experiments, especially for transformer-based models. One of the possible ways to improve can be to implement more classification heads for sentiment analysis, one with categorical labels prediction and one for the VAD space prediction. For instance, given that IEMOCAP offers more than one way to label emotions we could have exploited this knowledge to train on two tasks at the same time combining the losses to get more general representations from the embedders. It's worth noting that the labels are given by at least 3 different annotators and some of the samples don't have a majority vote among annotators (in our case, we decided to mask these samples), therefore the labels could be thought to be non mutually exclusive, and train one head with a multi-label classification task, in order to use the most of the data available.

8 Links to external resources

- Link to the Github repository of this project
- Link to the IEMOCAP dataset page

References

Abdul Badshah, Jamil Ahmad, Nasir Rahim, and Sung Baik. 2017. Speech emotion recognition from spectrograms with deep convolutional neural network. pages 1–5.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Soumya Dutta and Sriram Ganapathy. 2023. Hcam – hierarchical cross attention model for multi-modal emotion recognition.

Abhinav Joshi, Ashwani Bhat, Ayush Jain, Atin Singh, and Ashutosh Modi. 2022. COGMEN: COntextualized GNN based multimodal emotion recognitioN. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4148–4164, Seattle, United States. Association for Computational Linguistics.

Taewoon Kim and Piek Vossen. 2021. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *CoRR*, abs/2108.12009.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Navonil Majumder, Devamanyu Hazarika, Alexander F. Gelbukh, Erik Cambria, and Soujanya Poria. 2018. Multimodal sentiment analysis using hierarchical fusion with context modeling. *CoRR*, abs/1806.06228.