Multimodal Sentiment Analysis: Progress Report #1

Lucía Prado - Andrés Martínez
Javier Ríos - Pablo García

April 2024

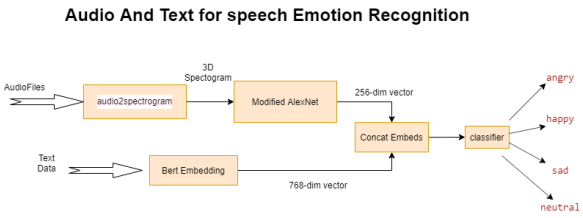
1 Problem and Hypotheses

In our study of Multimodal Sentiment Analysis, we identified several potential challenges to tackle. After careful consideration, we opted to focus on Multimodal Feature Fusion for Speech Emotion Recognition.

This problem stems from the need to address the limitations of models that rely solely on audio features to build effective classifiers. In our paper, we investigate a novel approach: a deep encoder model that exploits both text data and audio signals simultaneously to enhance the comprehension of speech data.

Given that emotional dialogue is composed both of sound and spoken content, we aim to encode the information from audio and text sequences using dual neural networks and then combine these sources to predict the emotion class. This architecture operates from the signal level to the language level, enabling comprehensive utilization of data information, in order to surpass models based only on audio characteristics. [3]

2 Mathematical and Algorithmic Approach



2.1 Text Processing

BERT (initialism for Bidirectional Encoder Representation from Transformers) is a model proposed in October 2018 in the paper *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. "Designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers" [2] By using context from both previous and following words, the resulting embeddings have a deeper sense of language context

BERT radically changes the prediction goal. Instead of predicting the next word in a sequence, it uses two strategies: Masked LM, which attempts to predict 15% of the words in each sequence, replacing them with a [MASK] token, and Next Sentence Prediction, learning to predict whether pairs of sentences are subsequent ones in the original document. This shift, though still provoking a conceptually simple model, has obtained new state-of-the-art results on NLP Tasks, overpowering far more complex models in them.

2.2 Audio Processing

2.2.1 Preprocessing

Before further processing we extract essential information about the audio signal:

- Sample rate: The frequency at which the audio signal is sampled (Hz).
- Audio data: The digital representation of the sound, captured by sampling the audio waveform at regular intervals determined by the sample rate.

In order to obtain the spectrograms from the audio data (WAV files), the next steps were followed:

- Audio to spectrogram (file path): extract sample rate and audio data using SciPy wavfile.read().
- Log spectrogram (audio, sample rate, window size, step size, eps): computes the log spectrogram and frequency axis of an audio signal using SciPy signal.spectrogram().
- 3D spectrogram (Sxx in, moments): converts a 2D spectrogram into a 3D representation by stacking the base spectrogram with its first and second-order deltas. It normalizes all three spectrograms, stacks them along the third dimension, and returns the resulting 3D spectrogram.

2.2.2 3D Spectrogram Processing

Once the audio is processed into a 3D spectrogram, we will pass it through a slightly modified version of the AlexNet architecture [4], in order to obtain an output vector that contains the information of the audio. In order to obtain a vector from the output tensor of the CNN, other transformations might be applied, in particular one of the proposed architectures is proceeding the AlexNet by an attention layer [4].

That being said, we expect the CNN to receive as an input an spectrogram of shape $f \times t \times c$ and return a processed 3D tensor of shape $F \times T \times C$, where F represents the frequency domain, T the time scale, and C the channel. We can consider the output as a

variable-length grid of L elements, $L = F \times T$. Each of the elements is a C-dimensional vector corresponding to a region of speech spectrogram, represented as a_i. Thus, we can represent the output as the following set:

$$A = \{a_1, \ldots, a_L\}, \quad \text{where} \quad a_i \in \mathbb{R}^C.$$

Intuitively, not all time-frequency units contribute equally to the emotion state of the whole utterance, i.e., not all the element vectors of set A contribute equally to the emotion state. Hence, the introduction of an attention layer, as a mechanism to extract the elements that are important to the emotion, after the CNN, becomes relevant. We use the following formulas to realize this idea:

$$e_i = \mathbf{u}^T \tanh(\mathbf{W}\mathbf{a}_i + \mathbf{b})$$
$$\alpha_i = \frac{\exp(\lambda e_i)}{\sum_{k=1}^{L} \exp(\lambda e_k)}$$
$$\mathbf{c} = \sum_{i=1}^{L} \alpha_i \mathbf{a}_i$$

With this definition, we can understand the CNN is followed by a MLP layer with the tanh as the non-linear activation. Then we measure the importance weight, e_i, of the a_i by the inner product between this new vector and the learnable vector $\mathbf{u}$. Lastly, α_i is calculated through the soft-max function, and the emotion vector c is computed as the weighted sum of set $\mathbf{A}$ with importance weights.

2.3 Classification

Once both types of input have been processed into their corresponding vectors, they are concatenated and passed through a final stage. This final processing will consist on a simple FC layer where the output will be a tensor representing the probability of the inputs belonging to each emotion.

3 Dataset

A great deal of thought and research has had to be devoted towards the choice of the data used to train the model. For now, we've focused on dealing with bimodal features (for instance, the model we've centered until this report uses only audio and text). Despite of this, a problem arose when we wanted to process the matching features of each modality

Given that the dataset is composed of dialogues between two people, there are different utterances which correspond to different sentiments. Even though the text was correctly segmented in the different utterances, in most the datasets we researched the audio wasn't. Therefore, a great deal of time had to be dedicated towards finding one that did segment it.

Finally, one was found, from the SAIL Lab at USC, originally used at this research [1]. Even though it has around 25GB of data (without taking into account the video parts, as we omitted those), we will take a feasible proportion of the data to train the models, in the order of the 100's of MB, at most.

4 Next Steps

If possible, we would like to study further the issues of Sentiment Discrepancy and Polarization: when dealing with data from different sources they might indicate opposite sentiments due to phenomena such as sarcasm.

References

[1] Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation 42* (2008), 335–359.

[2] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[3] Yoon, S., Byun, S., and Jung, K. Multimodal speech emotion recognition using audio and text. In *2018 IEEE spoken language technology workshop (SLT)* (2018), IEEE, pp. 112–118.

[4] Zhang, Y., Du, J., Wang, Z., Zhang, J., and Tu, Y. Attention based fully convolutional network for speech emotion recognition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (2018), IEEE, pp. 1771–1775.