INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Self-attention for Speech Emotion Recognition

Lorenzo Tarantino[1,2], *Philip N. Garner*[3], *Alexandros Lazaridis*[2]

[1]Ecole Polytechnique Federale de Lausanne, Switzerland
[2]Artificial Intelligence and Machine Learning Group, Swisscom
[3]Idiap Research Institute, Martigny, Switzerland

lorenzo.tarantino@daskell.com, phil.garner@idiap.ch, alexandros.lazaridis@swisscom.com

Abstract

Speech Emotion Recognition (SER) has been shown to benefit from many of the recent advances in deep learning, including recurrent based and attention based neural network architectures as well. Nevertheless, performance still falls short of that of humans. In this work, we investigate whether SER could benefit from the self-attention and global windowing of the transformer model. We show on the IEMOCAP database that this is indeed the case. Finally, we investigate whether using the distribution of, possibly conflicting, annotations in the training data, as soft targets could outperform a majority voting. We prove that this performance increases with the agreement level of the annotators.

Index Terms: speech emotion recognition, self-attention, global windowing

1. Introduction

In the provision of automated telephone services, we are generally interested in the words spoken by a client. However, the emotional state of the client is also important. It is an indicator of the client's general satisfaction with the process; it can also be a cue for the dialogue agent to switch to a human operator.

The field of Speech Emotion Recognition (SER) is concerned with the automatic detection of the emotional state of a person from spoken utterances. The SER problem has been addressed for several years using statistical methods and machine learning algorithms, such as Support Vector Machines (SVMs) and various regression algorithms. Over the last decade the available computational power has enabled the development of neural network based deep architectures such as the *attention mechanism*. Those algorithms, which are able to model more complex patterns within speech utterances, have led to more robust models for recognizing the emotional state of the speakers.

Bidirectional recurrent neural networks (BiRNNs) in the SER field were introduced by Lee et al. [1]. Subsequently, Chernykh et al. [2] built on top of BiRNNs a connectionist temporal classification (CTC) loss. In the following years several papers showed the contribution of attention models in the SER field. Among those, Mirasmadi et al. [3] approached the problem using local attention, a new weighted time-pooling algorithm that, instead of mean pooling over time, computes a weighted sum of the attention output (where the weights are learned within the model). Neumann et al. [4] used attention on top of a convolutional neural network, showing that convolutions could tackle the problem with similar performance. Ramet et al. [5] investigated several attention methods ([6, 7, 8, 9, 10]) on the SER task. They also proposed a new attention method applied to BiRNNs with LSTM cells using recurrent layers in the inner computation of the attention. This model is the current state of the art.

Although the attention mechanism combined with RNNs has improved performance on SER, it is limited by the state of the cell (e.g., LSTM) that can contain a limited amount of information; it is also affected by exploding and vanishing gradient problems [11]. For this reason we move from traditional attention mechanism to *self-attention*.

In audio processing, windowing is intended as very small and *local* windows in which it is assumed that the audio signal is constant. In this paper we propose a *global* windowing system that works on top of the previous one, applies windows of a larger order of magnitude and captures deeper relationships within the utterances leading to a better expressivity of the input. We propose two different downstream systems, one end-to-end attention learning from raw audio and another one based on a prior features extraction step, both trained with two methods: classification and regression. Moreover, to the best of our knowledge, *self-attention* (see section 2) has never been used before in the SER task. This leads to reducing training and inference times and to be able to better explain the behavior of the model given the attention weights. Finally, we show that this approach leads to state of the art results for weighted accuracy (WA) and unweighted accuracy (UA).

The paper is structured in the following way: in Section 2 we describe *self-attention* and its application on our SER task, the reasons behind the choice of different features set and the requirements that the model should respect. In Section 3, dataset, methods and model architecture are described. In Section 4, we present our results and we compare them to the previous state of the art.

2. Proposed Method

In the work of Vaswani et al. [12], it was shown how recurrent neural networks could be substituted with *self-attention*. A new attention technique based on an encoder-decoder structure that does not use any kind of recurrence, but instead uses weighted correlations between the elements of the input sequence, was introduced. The role of the encoder function is to map the input sequence into several attention matrices, while the decoder uses those matrices to generate a new token. We will focus only on the encoder part, since it is the one needed for the implementation of our proposed architecture. The *Transformer*, the model that uses *self-attention*, showed to be able to get state of the art results in translation tasks with one or two orders of magnitude (depending on the size of the model) lower computing cost with respect to time, than RNNs and in several other NLP tasks [13].

The concept behind the *Transformer* relies on the idea that each element of the input sequence can be projected through a linear function into three different representations of itself: a *query*, a *key* and a *value*. More formally, given an input se-

http://dx.doi.org/10.21437/Interspeech.2019-2822

quence $\boldsymbol{X}$:

$$q_i = w_q^T x_i, v_i = w_v^T x_i, k_i = w_k^T x_i \quad (1)$$

where x_i is the i_{th} element of $\boldsymbol{X}$ and w_q, w_v and w_k are the linear projections that map the i_{th} element to *query*, *value* and *key* respectively. *Query*, *value* and *key* have a dimensionality $1 \times d\_model$, where d_model is an hyperparameter.

In order to exploit matrix multiplications, attention is computed on the set of queries of the sequence:

$$z = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (2)$$

where *Q* is the set of queries, *K* is the set of keys, *V* is the set of values of the sentence and d_k is a scaling factor. The variable z obtained is the attentional matrix ($N \times d\_model$), where N is the number of elements contained in the input sequence. It is straightforward to see that this process can be repeated several times, stacking *self-attention* layers on top of each other.
A peculiarity of the *Transformer* is the *multi-headed attention*. *Heads* refer to the number of projections of each variable (*query*, *value* and *key*) applied to each element of the input.
In order to give the model the knowledge of the order within the sequence, it is required to add a positional encoding. This means summing a sinusoid function with a large period over the input before feeding it to the first encoder layer.

RNNs and attention models have drawbacks on SER which are mitigated by using *self-attention*. Indeed, emotions have long term correlations and it is known that RNNs have a decaying memory that is insufficient to preserve those correlations [11]. *Self-attention* sees every frame of the utterance simultaneously, so it cannot "forget the past". Moreover, each element of the input sequence is represented with $3 * n_{heads}$ projections, compared to only one representation as in the case of RNNs. This enables a better representation of the emotions as well as an increased expressivity of the model. Furthermore, the output of a traditional attention mechanism and of a *Transformer*'s encoder differ. The former is a matrix where each frame is weighted given its relevance for the task and the latter is a matrix where each frame is weighted with the learned correlations with the other frames. This latter approach adds more information about the actual value of the frame within the context.

Fixed length input sequences are required in order to apply the *Transformer*. Generally, windowing is used to extract features from raw audio with a small window (25ms) and small step size (10ms). Moreover, we used a sliding window of length W with a step size s (where s is a percentage of W) on top of the already extracted frames, as shown in Figure 1. The reason

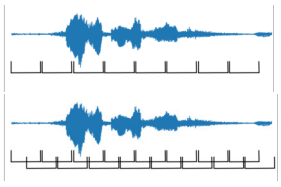

Figure 1: *Global sliding window with 0% (above) and 50% (below) overlapping.*

behind this *global* window is the fact that it can express relationships between datapoints that otherwise the model would have never been able to extract. We expect that a larger window size would lead to better results since the model would be able to see a larger context. Furthermore, a smaller step size leads to feeding the model more fixed length utterances with an $s * W$ offset difference between each other, resulting in an increased input representation.

3. Experimental setup

3.1. Input features

In this work we investigated not only the performance of a new model, but also the performance of the model given different inputs: our hypothesis is that learning the features extraction phase from raw audio will lead to detect task dependent features. For this reason, we want to create a fully end-to-end model that we hope that will be able to automatically extract meaningful features and compare it with a standard features extraction method. [14], [15] and [16] are only a few of the publications that show the effectiveness of convolution as features extraction layers from raw audio. A drawback of this approach is that by adding the features extraction task to the model pipeline additional complexity is added to the end to end system increasing the need in the number of datapoints for training a robust model. The relatively small amount of training data in our case, only 5.5 hours of speech, could lead to a partial learning of the input representation.
As for the engineered features, we evaluated our methodology on the IS09 [17] features set (384 features) because it is a common set used for SER tasks and it has been used by [5] to get the latest state of the art results. Even if it is not been used as extensively as IS09, we extracted also the eGeMaps set [18]: this set showed to be a good substitute of IS09 in several works, such as [19], [20] and [21]. The eGeMaps set is a good trade-off between expressivity of the input and training times, since it contains only 88 features that means shorter computational times with an inferior representation of the frames. Both features set were extracted using the openSMILE framework [22] with a window size of 25ms and step size of 10ms.

3.2. Database

IEMOCAP database [23] was chosen for our experiments since it has been established in the literature on the SER field as a benchmark. Moreover, it contains high frequency recording audio data (16kHz sample rate), both genders, 9 emotions and improvised and scripted speech, that the literature showed to have different complexity when making inference [4], [5]. Out of the 9 emotions we focused on four of them (*angry*, *happy*, *neutral* and *sad*) in order to have comparable results with previous research. We trained and evaluated the model a second time without modifying the structure of the model and substituting the *happy* class with the *excitement* one in order to compare our results with [2].

Each utterance is labeled by three to four annotators and the classification label is the majority label between the annotations. We investigated the distribution of the four emotions used given different levels of agreement of the annotators. As we can see in Figure 2, the dataset is very imbalanced and the more precision we require on the majority label, the more the *happy* class shrinks.

Since understanding the quality of the dataset is essential to create a well-performing model, we have studied the distribution of annotations given the majority label. In Figure 3 we plotted the distribution of the annotations of the four classes for

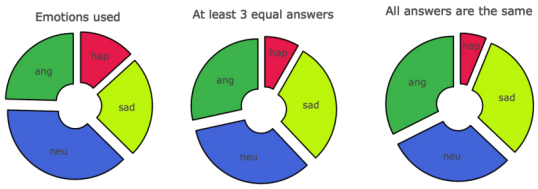

Figure 2: *Different distributions of the four emotions used depending on the quality of the annotations.*

each class. We notice that every class has around 75% of annotations corresponding to itself except for the happy class, that has only 68%. This result explains the "shrinking" in Figure 2. This means that the dataset, labeled with the majority annotations, contains 25% or more of noise and that each sample cannot be described with only one emotion.

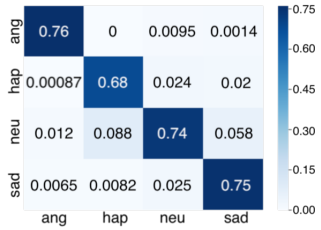

Figure 3: *Distribution of the annotations of angry, happy, neutral and sad given the majority label.*

3.3. Classification or Regression

A well known issue in the SER field is that very often the annotators are not unanimous in classifying the utterances, as highlighted by Lotfian et al. in [24]. That is understandable since it is rare to express only one emotion when speaking. For this reason, we worked on two different methodologies: classification and regression.
Classification labels are the majority labels between the annotations of one utterance (categories). Regression targets, on the contrary, do not consider the majority label, but, instead, the proportion of the classes within the annotations (scalars). Regression targets are calculated as proportion between the four classes only. For instance, if an utterance has four annotations that are [happy, happy, angry, excited], the target results will be angry 0.25, happy 0.5, neutral 0, sad 0. Following the previous example, the label for classification would have been happy. This *soft labeling* given by the regression targets represents a closer distribution of the utterances' emotional content to the real one. Our assumption is that a better distribution of the labels will lead to a better learning process of the model.
In order to remain consistent with the results between the two methodologies, we used the same data for classification and regression, even if for regression we would have not needed to filter out the utterances with inconsistencies (that is with no majority label).

3.4. Normalization of the input

We normalized engineered features (IS09 and eGeMaps) subtracting the mean and dividing by the standard deviation of neutral features as proposed in [5]. For raw audio signals, we subtracted the mean and divided by the standard deviation of the signal in order to have a final mean around 0 and a final standard deviation equal to 1.

$$normalized\_x = \frac{x - mean_{x\_neutral}}{std_{x\_neutral}} \quad (3)$$

$$normalized\_signal = \frac{signal - mean_{signal}}{std_{signal}} \quad (4)$$

3.5. Model architecture

When the input is the signal itself, the first layers of the model are convolutions which role is to extract low level features. These layers are made of 6 stacked one-dimensional convolutions and max pooling layers plus a final linear layer that maps the output of the convolution to the dimension of the model. More formally, an input signal of shape $1 \times W$ is mapped first to a matrix of shape $N \times dim$ and then to $N \times d\_model$ by the linear layer, where N and dim are dependent by the kernel size, stride and number of convolutions. When the input is a feature set ($N \times dim$), a linear layer maps each frame from dim to d_model.
In both cases, the *self-attention* layers will receive as input a matrix $N \times d\_model$. As explained in section 2, *self-attention* layers produce an attentional matrix: each row of the matrix represents a frame in the context of all the other frames of the input utterance. In order to extract even deeper relationship between frames through time and to reduce the dimensionality of the output, two-dimensional convolutional layers with max pooling are applied on top of the attentional matrix and a final linear layer maps the output to the number of classes.

3.6. Windowing

In order to prove our assumption that more overlapping (smaller step size) means a better representation of the input, we repeated training and evaluation for step sizes 0.1, 0.2, 0.3, 0.5 and 1 (no overlapping) on five window sizes (100, 200, 300, 400 and 500 frames). As we can see from Figure 4, independent of the window size, a smaller step size leads to a smaller loss. The colored line represent the mean of the experiments, while the colored area is the standard deviation computed on the five experiments for each step size.

3.7. Aggregation method

The model outputs four probabilities for every chunk produced by the windowing process. Since our final goal is to give a prediction for the original utterance, we aggregate the probabilities originated from the same sequence doing a simple average and using the highest one as the class prediction.

Confident of these results, we used for all our experiments a window size of 500 frames and a step size of 50 frames ($s = 0.1$).
Since the dataset is very unbalanced, especially for the happy class, we used a weighted loss, where the weights were computed starting from the number of samples obtained after windowing.

4. Experimental Results

4.1. Results comparison

We used a 5-fold cross validation for each combination of input types, i.e. IS09, eGeMaps and raw audio and methodologies, i.e. classification and regression.

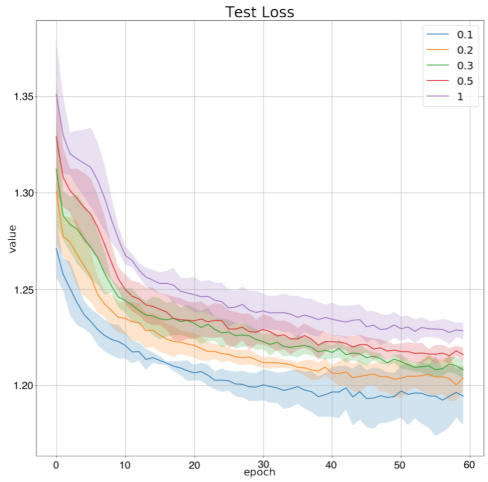

Figure 4: *Mean and standard deviation of the test (5th session) loss for different step sizes given window sizes of 100, 200, 300, 400 and 500 frames.*

Table 1: *Performances of previous and our methods on the whole dataset with a 5 fold cross validation.*

Method	WA	UA
Neumann et al. [4]	56.1	-
Ramet et al. [5]	62.5	59.6
IS09 - classification	**68.1**	**63.8**
IS09 - regression	66.6	62.3
eGeMaps - classification	65.0	60.6
eGeMaps - regression	64.6	58.5
Raw audio - classification	64.0	57.4
Raw audio - regression	65.4	58.5

As we can see in Table 1, each method got better results than previous state of the art: this shows that the model itself is robust on different kinds of input. The IS09 set is the input that performed best, getting a 5.6% and 9% absolute and relative improvement on the WA and a 4.2% and 7% absolute and relative improvement on the UA. As expected, eGeMaps features did not perform as well as the IS09 given their reduced dimensionality.
Raw audio is the only input that performed better when the task was a regression instead of a classification. This confirms our observation that having a good distribution of the emotions over the utterance labels means learning a features representation of the input that is closer to the real one. We report also the results for the improvised dataset and gender in Table 2. As shown by [4] and [5], the improvised dataset is simpler to classify. This is due to the naturalness of the actors when they are playing, who give more weight on the emotional content rather than what they are saying. It is also curious to observe that it is easier to predict women's emotions than men's ones.

4.2. Annotations agreement level

As already stated, annotators often gave different answers for the same utterance. We want to understand the performance of our model with respect to different consistencies in the majority labels. The model performs better as the consistency of the answers increases as shown in Table 3. As expected, the better the labeling, the better the model can perform.

Table 2: *Performances of our model for the two genders, improvised and scripted sessions using classification and IS09 features set.*

Type/Gender	WA	UA
M	65.71	59.59
F	69.59	65.21
Scripted	64.59	50.12
Improvised	**70.17**	**70.85**
Improvised Neumann et al. [4]	62.1	-
Improvised Ramet et al. [5]	68.8	63.7

Table 3: *Performances of our model for different consistencies using classification and IS09 features set.*

Agreement level	WA	UA
2 equal answers	59.43	58.01
3 equal answers	75.59	73.97
4 equal answers	77.57	82.24

4.3. The excitement class

As can be seen in Figure 2 the amount of training data in the *happy* class is relatively small in comparison with any of the other three classes used in all our experiments. Due to this fact, in our final experiment, the *happy* class was replaced with the *excitement* class for having a more balanced distribution of training data across emotions. The structure of the model was kept unchanged and the IS09 features set was used as well. In Table 4, we can see that our model achieves better results than the ones of the previous task, outperforms previous work and it starts to become comparable to human performances.

Table 4: *Performances of [2] and our methods with a 5 fold cross validation using the excitement instead of the happy class.*

Method	WA	UA
Chernykh et al. [2]	54.0	54.0
Human performances [2]	69.0	70.0
IS09 - classification	**64.33**	**64.79**

5. Conclusions

In this paper, we demonstrated the effectiveness of a new windowing system that is able to capture hidden relationships within the data, improving the performances for SER. This technique combined with *self-attention* outperforms the previous state of the art on the IEMOCAP dataset with respect to WA and UA. Moreover, we conducted a study over the possible input representations showing that the performances of the IS09 features set are better than eGeMaps and learned features through convolutions. Finally, we showed that a better distribution of the emotions over the labels is necessary to learn features from raw audio and that the performances of our model increases proportionally to the agreement level of the annotators.

6. References

[1] J. Lee and I. Tashev, "High-level feature representation using recurrent neural network for speech emotion recognition," in *INTERSPEECH*, 2015.

[2] V. Chernykh, G. Sterling, and P. Prihodko, "Emotion recognition from speech with recurrent neural networks." *CoRR*, vol. abs/1701.08071, 2017.

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

[4] M. Neumann and T. Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," 06 2017.

[5] G. Ramet, P. N. Garner, M. Baeriswyl, and A. Lazaridis, "Context-aware attention mechanism for speech emotion recognition," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 126–131, 2018.

[6] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *ACL*, 2016.

[7] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *HLT-NAACL*, 2016.

[8] W. Pei, T. Baltrusaitis, D. M. J. Tax, and L.-P. Morency, "Temporal attention-gated model for robust sequence classification," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 820–829, 2017.

[9] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Attention and localization based on a deep convolutional recurrent model for weakly supervised audio tagging," in *INTERSPEECH*, 2017.

[10] T. Zhang, G. Lin, J. Cai, T. Shen, C. Shen, and A. C. Kot, "Decoupled spatial neural attention for weakly supervised semantic segmentation," *CoRR*, vol. abs/1803.02563, 2018.

[11] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[14] D. Bertero, F. B. Siddique, C.-S. Wu, Y. Wan, R. H. Y. Chan, and P. Fung, "Real-time speech emotion and sentiment recognition for interactive dialogue systems," in *EMNLP*, 2016.

[15] D. Palaz, "Towards end-to-end speech recognition," *EPFL thesis no 7054, in collaboration with IDIAP Research Institute*, 2016.

[16] P. Tzirakis, J. Zhang, and B. W. Schuller, "End-to-end speech emotion recognition using deep neural networks," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5089–5093, 2018.

[17] B. Schuller, S. Steidl, and A. Batliner, "The interspeech 2009 emotion challenge," 01 2009, pp. 312–315.

[18] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Devillers, J. Epps, P. Laukka, 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, pp. 190–202, 2016.

[19] L. Chao, J. Tao, M. Yang, Y. F. Li, and Z. Wen, "Long short term memory recurrent neural network based multimodal dimensional emotion recognition," in *AVEC@ACM Multimedia*, 2015.

[20] F. Ringeval, B. W. Schuller, M. F. Valstar, J. Gratch, R. Cowie, S. Scherer, S. Mozgai, N. Cummins, M. Schmitt, and M. Pantic, "Avec 2017: Real-life depression, and affect recognition workshop and challenge," in *AVEC@ACM Multimedia*, 2017.

[21] E. Marchi, B. W. Schuller, S. Baron-Cohen, O. Golan, S. Bölte, P. Arora, and R. Häb-Umbach, "Typicality and emotion in the voice of children with autism spectrum condition: evidence across three languages," in *INTERSPEECH*, 2015.

[22] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: ACM, 2013, pp. 835–838. [Online]. Available: http://doi.acm.org/10.1145/2502081.2502224

[23] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. M. Provost, S. Kim, J. N. Chang, S. Lee, and S. Narayanan, "Iemocap: interactive emotional dyadic motion capture database," *Language Resources and Evaluation*, vol. 42, pp. 335–359, 2008.

[24] R. Lotfian and C. Busso, "Curriculum learning for speech emotion recognition from crowdsourced labels," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.