An Efficient Temporal Modeling Approach for Speech Emotion Recognition by Mapping Varied Duration Sentences into Fixed Number of Chunks

Wei-Cheng Lin and Carlos Busso
Multimodal Signal Processing (MSP) lab, Department of Electrical and Computer Engineering
The University of Texas at Dallas, Richardson TX 75080, USA
weichi-eng.lin@uta.edu, busso@utdallas.edu

Abstract
Speech emotion recognition (SER) plays an important role in multiple fields such as healthcare, human-computer interaction (HCI), and security and defense. Emotional labels are often annotated at the sentence-level (i.e., one label per sentence), resulting in a sequence-to-one recognition problem. Traditionally, studies have relied on statistical descriptions, which are computed over time from low level descriptors (LLDs), creating a fixed dimension sentence-level feature representation regardless of the duration of the sentence. However, sentence-level features lack temporal information, which limits the performance of SER systems. Recently, new deep learning architectures have been proposed to model temporal data. An important question is how to extract emotion-relevant features with temporal information. This study proposes a novel data processing approach that extracts a fixed number of small chunks over sentences of different durations by changing the overlap between these chunks. The approach is flexible, providing an ideal framework to combine gated network or attention mechanisms with long short-term memory (LSTM) networks. The experimental results based on the MSP-Podcast dataset demonstrate that the proposed method not only significantly improves recognition accuracy over alternative temporal-based models relying on LSTM, but also leads to computational efficiency.

1. Introduction
Detecting human emotion state via speech is helpful in multiple fields such as human-computer interaction (HCI) [1] or healthcare [2, 3]. Therefore, speech emotion recognition (SER) has become a popular research area. Most existing corpora are labeled at the sentence-level, where one global label is assigned per sentence [4–6]. With these labels, SER is often formulated as a sequence-to-one problem (i.e., mapping sequence of frames into a single label). Traditional methods deal with this problem by using high level descriptors (HLDs) such as mean, minimum and variance, estimated from low level descriptors (LLDs) extracted from speech (e.g., fundamental frequency, Mel frequency cepstral coefficients (MFCCs), energy). This approach generates a fixed dimension vector for a sentence, regardless of its duration, reformulating the problem as a static one-to-one machine learning problem. However, HLDs are not able to reflect the dynamic temporal information in the expression of emotion, leading to limited performance for SER systems. Therefore, recent studies have explored methods that can directly build SER systems from frame-based features, without relying on predefined functionals used in HLDs.

Deep learning approaches for SER systems have recently led to state of the art performance [7]. Different architectures exploring temporal information such as recurrent neural network (RNN), convolution neural network (CNN) or hybrid neural network (CNN-LSTM) have shown state-of-the-art performance by deriving features directly from LLDs or raw waveform [8–12]. These temporal models can be divided into two main categories for dealing with sentences with different lengths. The first category formulates the sequence-to-one task as a sequence-to-sequence task by relying on methods such as the connectionist temporal classification (CTC) loss [8, 9] and Markov chain-based approaches [13]. These approaches aim to create latent frame-by-frame variables for the emotional labels, which are then used to train the model. Although this method can effectively get rid of non-emotional frames, the emotional frames are labeled with the same class as the label assigned to the entire sentence. Moreover, a big issue when using RNN-based models in practical applications is the computational resources needed for sequences with long duration [14]. The second category uses deep learning models to extract sentence-level feature representations directly from the data, avoiding extracting predefined HLDs. After the feature representation is extracted, the problem is formulated as a one-to-one task [10, 11, 15]. These methodologies jointly learn feature extraction and build the SER models, resulting in better performance compared to traditional HLD representation. Nevertheless, they have some limitations such as requiring sentences with fixed length [11, 12], or using temporal-based pooling before the output layer [10, 15]. These approaches either truncate a sentence, append zeros or ignore primitive temporal information.

This paper proposes a novel and flexible data processing approach to model temporal acoustic information that addresses some of the key limitations of current temporal formulations in SER. The idea of the approach is to split sentences of different durations into a fixed number of small chunks by adjusting the overlap between chunks. This chunking procedure obtains a fixed number of data chunks, regardless of the duration of the sentence. It does not require dropping frames or appending zeros, preserving the complete temporal information of the original input sequence. The chunks are independently processed by a long short-term memory (LSTM) network with shared parameters. A key advantage of the approach is the flexibility offered by this formulation to combine the fixed number of feature representations created by the LSTMs for each of the chunks. This study proposes to combine these representations with a mean pooling layer (NonAtten network), a gated network (GateVec network) or an attention mechanism (AttenVec network). Finally, we obtain a sentence-level attention feature representation to generate emotion predictions via a fully connected output layer. Another key advantage of this approach is that it can significantly reduce the number of time steps by splitting the data sequence into small chunks, improving the computational efficiency of the architecture.

We evaluate the proposed models on the MSP-Podcast dataset [4], formulating the problem as a regression task to predict emotional attributes. Evaluating the models using concor-
els. MTL has been implemented in SER by considering several valence and dominance with a single model using the number of chunks is fixed. Contrast to previous studies, our formulation constructs attention by building an attention model using frame-level features. Typ-chunks is through attention models, which has been widely used chunk as a function of the duration of the utterances. This key approach generates data chunks by varying the step size of the size of the chunks as a fixed number. However, our proposed feature representation in the network. These studies set the step overlap between chunks) can increase the discrimination of the put data chunks. They found that smaller step size (i.e., more global features from LLDs, which obtains better performance as the sentence-level feature representation of a static classi-
case curves with the probabilities for the emotions. Finally, chunk, where the outputs of the model were used to estimate a level feature by stacking the neighboring LLD frames to train level feature for SER. Han et al. form their segment-
the prediction accuracy, but also the computational efficiency.

2. Related Work

Various studies have utilized the concept of chunk or segment-
level feature for SER. Han et al. [16] formed their segment-
level feature by stacking the neighboring LLD frames to train a deep neural network (DNN). They train a DNN over the chunk, where the outputs of the model were used to estimate the probability of each emotion for that frame. This approach created curves with the probabilities for the emotions. Finally, they estimated statistics over these curves, which were used as the sentence-level feature representation of a static classi-
er with extreme learning machine (ELM). Tzinis and Potamianos [17] employed HLD to represent segment-wise global features from LLDs, which obtains better performance compare to LLDs under a LSTM model. Tarantino et al. [18] built a self-attention model by setting different step size of input data chunks. They found that smaller step size (i.e., more overlap between chunks) can increase the discrimination of the feature representation in the network. These studies set the step size of the chunks as a fixed number. However, our proposed approach generates data chunks by varying the step size of the chunk as a function of the duration of the utterances. This key distinction leads our models to achieve clear improvements.

In our study, one of the proposed approaches for fusing the chunks is through attention models, which has been widely used in SER [19–22]. The most common way to apply this method is by building an attention model using frame-level features. Typically, the inputs of the attention model are activations of intermediate representation layers in the network. This approach produces attention weights per frame, which are used to obtain a final attention vector to recognize emotions [20–22]. In contrast to previous studies, our formulation constructs attention model at the chunk-level, reducing the computational cost since the number of chunks is fixed.

Another feature of our models is that we recognize arousal, valence and dominance with a single model using multitask learning (MTL). MTL allows a model to learn a common feature representation (i.e., a shared layer) by solving multiple related tasks, which increases the generalization of the models. MTL has been implemented in SER by considering several tasks, including multiple emotional attribute prediction [23,24], gender classification [25], and primary and secondary categori-
el emotion classification [26]. Since previous studies have consis-
tently showed better performance by using multitask learning, we implement our model with MTL.

The main contributions of this paper is a novel chunk-based temporal modeling method that can map varied length data into fixed number of data chunks. This novel formulation is flexi-
ble, allowing different feature extraction methods and different combinations of the chunk-level representation. Our proposed chunk-based temporal modeling not only increases the accuracy of our predictions, but also reduces the complexity of model.

3. Proposed Methodology

3.1. Chunk-Based Segmentation

The core idea of our chunk-based segmentation method is to split a varied duration sentence into a fixed number of data chunks that have the same fixed duration. We achieve this goal by changing the step size (i.e., overlap between chunks). First, we need to set our desired length for the chunk window \( w_c \). This variable should be big enough so the models can estimate reliable emotional information from the chunk, and small enough to be able to process short sentences. We estimate the maximum sentence duration \( T_{\text{max}} = \max\{T_1, T_2, \ldots, T_i, \ldots, T_N\} \), where \( T_i \) denotes the duration of sentence \( i \). We use \( T_{\text{max}} \) to estimate a fixed number of chunks \( C \), according to:

\[
C = \left\lceil \frac{T_{\text{max}}}{w_c} \right\rceil.
\]

Since the overlap between chunks for longer sentences will be limited with this approach, we could increase the value of \( C \) by, for example, multiplying this value by an integer \( n \) (e.g., \( nC \)). The step size of the chunks \( \Delta c_i \) for sentence \( i \) is given by Equation 2. This equation shows that as we increase \( C \) (i.e., the number of chunks), \( \Delta c_i \) decreases, resulting in more overlap between chunks.

\[
\Delta c_i = \frac{T_i - w_c}{C - 1} \quad (2)
\]

Figure 1 visualizes the proposed approach for two sentences with different durations. The key difference between them is the chunk step size \( \Delta c_i \) (the overlapping area between chunks). By adjusting the chunk step size, this approach is able to split different duration sentences into a fixed number of chunks \( C \) that have the same duration \( w_c \). Section 4.2 describes the actual implementation of the approach, including the values for \( w_c \).

A key advantage of formulating SER problems with the proposed chunk-based segmentation is the simplification in modeling temporal information. Two important steps are (1) extracting feature representation from speech, and (2) combining chunk-based feature representations. The next two subsections describe these steps.

3.2. Extracting Feature Representation

Each chunk has a fixed length so extracting a feature repre-
sentation is straightforward \( (w_c) \). As shown in Figure 2, this study restricts the analysis to LSTMs, although several alter-
native methods can be used (e.g., CNNs, estimation of HLDs per chunk). We extract the feature set proposed for the Inter-
phasia 2013 computational paralinguistics challenge [27] us-
ing the OpenSmile toolkit [28]. The extracted LLDs include spectral, prosodic and energy-based acoustic features such as the fundamental frequency \( (f_0) \), energy, and MFCCs. In total, the set includes 130 frame-based acoustic features, which are normalized by subtracting the mean and dividing by the stan-
dard deviation (these parameters are estimated over the training.
set). Notice that we do not extract HLDs. The normalized input LLDs (X) are first split into data chunks \( \{X_1, X_2, \ldots, X_C\} \) where \( X_i \in \mathbb{R}^{m \times d} \). The dimension \( m \) is the number of frames per chunk and \( d \) is the dimension of the feature vector (i.e., \( d = 130 \)). Then, we feed these data chunks into two consecutive LSTM shared layers with \( b \) hidden nodes. We exploit the final time step of the output as the representation vector for each chunk, denoted as \( \langle h_1, h_2, \ldots, h_C \rangle \) where \( h_i \in \mathbb{R}^{1 \times b} \).

### 3.3. Combining Chunk-Based Feature Representations

Having a fixed number of chunks per sentence regardless of its duration simplifies the aggregation of temporal information across different chunks to form a sentence-level feature representation. Our formulation is flexible, where several approaches can be used. This study explores three alternative methods illustrated in Figure 2.

**NonAtten Model – Fig. 2(a):** After we obtained the chunk-level representations \( \{h_1, h_2, \ldots, h_C\} \), we directly average these vectors to obtain the sentence-level representation.

**GatedVec Model – Fig. 2(b):** The gated mechanism [29] allows the model to control the information flow from different channels. Equation 3 shows this operation, which consists of a sigmoid neural network (NN) layer \( (W_e, b_e) \) and a pointwise multiplication operation. By concatenating the gate model after the LSTM shared layer, we can produce the gating weights \( g_{i,e} \) (scalar) for each emotional attribute \( e \), where \( e \in \{aro, dom, val\} \). This approach obtains the sentence-level representation vector \( s_a \) with equation 4.

\[
g_{i,e} = \sigma(W_e \cdot h_i + b_e) \quad (3)
\]

\[
s_a = \sum_{i=1}^{C} g_{i,e} h_i \quad (4)
\]

**AttenVec Model – Fig. 2(c):** We first stack \( \{h_1, h_2, \ldots, h_C\} \) into a chunk-level feature map \( H \in \mathbb{R}^{C \times b} \) and feed \( H \) into a vanilla RNN attention model. Then, we train the attention weights \( \alpha_{i,e} \) for each emotional attribute \( e \) (i.e., different attention model for different emotional attributes) by using the general function from Luong et al. [30]. We use these attention scores to multiply the corresponding time step’s hidden state \( \{\overrightarrow{h}_{1,e}, \overrightarrow{h}_{2,e}, \ldots, \overrightarrow{h}_{C,e}\} \), where \( \overrightarrow{h}_{i,e} \in \mathbb{R}^{1 \times q} \). The dimension \( q \) is the number of nodes in the RNN attention model. This approach results in the context vector \( c_e \) (Eq. 5). Finally, we concatenate the vector \( c_e \) with the last hidden state \( \overrightarrow{h}_{C,e} \), passing through a NN layer \( (W_e) \) with the tanh activation function to obtain a sentence-level feature representation \( s_a \) (Eq. 6). Since the time steps in the RNN layer is fixed to \( C \) (i.e., attention on chunks rather than attention on all the input frames), our attention model is very computationally efficient.

\[
c_e = \sum_{i=1}^{C} \alpha_{i,e} \overrightarrow{h}_{i,e} \quad (5)
\]

\[
s_a = \tanh(W_e [c_e; \overrightarrow{h}_{C,e}]) \quad (6)
\]

For the three models in Figure 2, we feed the sentence-level feature representation \( s_a \) into their corresponding emotional attribute output layer, which is formed by two fully connected layers. The outputs of the multitask frameworks are the predictions for arousal, valence, and dominance. We do not fine-tune the hyperparameters for the multitask model in this work. Our loss function \( L_{total} \) is just a direct summation of the different task losses: \( L_{total} = L_{aro} + L_{dom} + L_{val} \).

### 4. Experimental Results

#### 4.1. Resources

We utilize the version 1.6 of the MSP-Podcast corpus [4] to build and evaluate our proposed approach. The dataset consists of spontaneous, emotional-rich speech segments collected from various online audio-sharing websites. Following the ideas presented in Marirood et al. [31], we process the segments to identify clean audio, from a single speaker, without music in the background. The dataset provides both categorical and attribute-based emotional annotations, which are labeled by at least five annotators for each speech segment using a crowdsourcing approach [32]. We build our multitask model for arousal (calm versus active), valence (negative versus positive) and dominance (weak versus strong), where the ground truth label is the average of the scores across annotators. The version 1.6 of the corpus is split into train (34,280 speech turns), development (5,958 speech turns) and test (10,124 speech turns) partitions. The partitions are defined to reduce cases where data from one subject is included in more than one set. The readers are referred to Lotfian and Busso [4] for more details.

---

### Table 1: Performance in CCC achieved by our proposed models, which are compared with LSTM-based models (i.e., LSTM(130) and LSTM(260)).

<table>
<thead>
<tr>
<th>Model</th>
<th>Aro [CCC]</th>
<th>Dom [CCC]</th>
<th>Val [CCC]</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM(130)</td>
<td>0.6520</td>
<td>0.5711</td>
<td>0.2031</td>
</tr>
<tr>
<td>LSTM(260)</td>
<td>0.6875</td>
<td>0.6045</td>
<td>0.2847</td>
</tr>
<tr>
<td>NonAtten</td>
<td>0.6781</td>
<td>0.6019</td>
<td>0.2925</td>
</tr>
<tr>
<td>GatedVec</td>
<td>0.6747</td>
<td>0.5944</td>
<td>0.3199</td>
</tr>
<tr>
<td>AttenVec</td>
<td>0.6947</td>
<td>0.6132</td>
<td>0.3072</td>
</tr>
</tbody>
</table>

### Table 2: The efficiency of the models in terms of number of parameters, Mega FLOPs, time cost for training and time cost for online processing.

<table>
<thead>
<tr>
<th>Model</th>
<th># of Par. [10^6]</th>
<th>MFLOPs</th>
<th>Train [sec/epoch]</th>
<th>Online [ms/utt]</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM(130)</td>
<td>0.323</td>
<td>5.67</td>
<td>437.1</td>
<td>547.5</td>
</tr>
<tr>
<td>LSTM(260)</td>
<td>1.052</td>
<td>18.49</td>
<td>439.4</td>
<td>598.1</td>
</tr>
<tr>
<td>NonAtten</td>
<td>0.323</td>
<td>0.49</td>
<td>74.9</td>
<td>42.2</td>
</tr>
<tr>
<td>GatedVec</td>
<td>0.324</td>
<td>0.49</td>
<td>246.6</td>
<td>44.6</td>
</tr>
<tr>
<td>AttenVec</td>
<td>0.577</td>
<td>1.50</td>
<td>353.1</td>
<td>45.6</td>
</tr>
</tbody>
</table>
This result shows the performance and number of parameters as the CCC valence improving the predictions from to the is the use of the chunk-based segmentation. Table 1 shows ing (i.e., average in millisecond per utterance during inference).

of parameters, megaFLOPs, time cost for training (i.e., average (from 323K to 324K), and without sacrificing computation cost (0.49 megAFLOPs). This solution is suitable for memory-limited or efficiency-oriented systems. The AttenVec model achieves the best CCC performance for arousal (CCC = 0.6947) and dominance (CCC = 0.6132), and a very competitive performance for valence (CCC = 0.3072). Valence is an attribute that is particularly challenging to predict with acoustic features [34]. The model has a reasonable increase in the number of parameters (from 323K to 577K parameters) and computation efficiency (from 0.49 to 1.50 MFLOP). These results show that the AttenVec model provides the best tradeoff between complexity and accuracy. We also increase the model complexity of our baseline LSTM model by doubling the number of its nodes (i.e., LSTM(260)). Tables 1 and 2 show that this model improves the recognition accuracy, but at a very expensive cost: approximately 3 times the number of parameters and MFLOPs of the LSTM(130) model. This model also increases the time cost for training and online processing. Even with these extra costs, the LSTM(260) model is still less accurate than the AttenVec model.

We evaluate the performance of the systems as we increase the duration of the sentence, which will result in less overlap between the chunks. For this analysis, we split the testing set into short (<5sec), middle (5-8sec) and long (>8sec) sentences. The test set has 4,280 short, 3,684 middle and 2,160 long sentences. Table 3 shows the results. The performance of the LSTM(130) model degrades as we increase the duration of the sentence for all three emotional attributes, showing poor accuracy for long sequences. In contrast, we can observe that the proposed chunk-based segmentation models systematically improve the performance for different duration of the data, especially for middle and long sequences. These results demonstrate that temporal modeling based on smaller chunks can be useful to aggregate long-term temporal information, leading to robust prediction accuracy, regardless of the duration of the sentences.

4.2. Experimental Settings

We implement our approach using chunks of 1 sec (ωc = 1). We have found that emotion can be estimated even with 0.5 secs speech segments [33], so 1 sec is a reasonable value. Since the duration of the sentences is between 2.75 and 11 secs, Tmax is 11 secs. The number of chunks is 11 (C = 11) with these parameters (Eq. 1). The window analysis for the LLDs is 32ms with a step size of 16ms (50% overlap). Therefore, the number of frames within one chunk is m = 62. The value for the step size for the chunks (∆ct) depends on the duration of the sentence (Ti) according to Equation 2. For example, if Ti = 6 secs, then ∆ct = 0.5 secs. For the network settings, we fixed the number of nodes in the layer matching the dimensions of the input (i.e., d = b = q = 130). We use dropout with p = 0.5 for the LSTM layers. We use batch normalization after the shared LSTM layers. We use Adam optimizer with a batch size of 128.

Our baseline model is directly trained with the entire input LLD sequence. We zero pad the feature vector until matching the maximum frame length of the dataset for batch training. We implement the LSTMs with either 130 nodes (LSTM(130)) or 260 nodes (LSTM(260)) per layer. The feature representation to estimate the output layers is the last frame of the LSTM layer.

4.3. Results and Analysis

Table 1 summarizes the recognition performance in terms of CCC. Table 2 shows the model efficiency in terms of the number of parameters, megaFLOPs, time cost for training (i.e., average in seconds per training epoch), and time cost for online processing (i.e., average in millisecond per utterance during inference).

The model provides the best tradeoff between complexity and efficiency advantages of our chunk-based segmentation.

We can further improve the recognition accuracy by adding gate (GatedVec) or attention (AttenVec) models instead of applying a mean pooling layer after the LSTM shared layer. The GatedVec model improves the prediction of valence with a minimal increase in the number of parameters (from 323K to 324K), and without sacrificing computation cost (0.49 megAFLOPs). This solution is suitable for memory-limited or efficiency-oriented systems. The AttenVec model achieves the best CCC performance for arousal (CCC = 0.6947) and dominance (CCC = 0.6132), and a very competitive performance for valence (CCC = 0.3072). Valence is an attribute that is particularly challenging to predict with acoustic features [34]. The model has a reasonable increase in the number of parameters (from 323K to 577K parameters) and computation efficiency (from 0.49 to 1.50 MFLOP). These results show that the AttenVec model provides the best tradeoff between complexity and accuracy. We also increase the model complexity of our baseline LSTM model by doubling the number of its nodes (i.e., LSTM(260)). Tables 1 and 2 show that this model improves the recognition accuracy, but at a very expensive cost: approximately 3 times the number of parameters and MFLOPs of the LSTM(130) model. This model also increases the time cost for training and online processing. Even with these extra costs, the LSTM(260) model is still less accurate than the AttenVec model.

We evaluate the performance of the systems as we increase the duration of the sentence, which will result in less overlap between the chunks. For this analysis, we split the testing set into short (<5sec), middle (5-8sec) and long (>8sec) sentences. The test set has 4,280 short, 3,684 middle and 2,160 long sentences. Table 3 shows the results. The performance of the LSTM(130) model degrades as we increase the duration of the sentence for all three emotional attributes, showing poor accuracy for long sequences. In contrast, we can observe that the proposed chunk-based segmentation models systematically improve the performance for different duration of the data, especially for middle and long sequences. These results demonstrate that temporal modeling based on smaller chunks can be useful to aggregate long-term temporal information, leading to robust prediction accuracy, regardless of the duration of the sentences.

5. Conclusions

This study proposed a novel segmentation approach that splits a sentence into a fixed number of chunks, which have the same duration. By changing the step size between chunks, we can process sentences of different durations. The experimental evaluation showed the benefits in efficiency and accuracy by using the proposed chunk-level temporal modeling methodology. This simple concept, which can be easily implemented, offers the flexibility to explore different feature representation (fixed size of the chunk) and different fusions for chunk-based representation (fixed number of chunks). This solution also facilitates parallel processing for GPU. We expect that this approach can be also effective in other sequence-to-one problems, beyond the field of affective computing.

Our future directions are: to (i) validate the proposed chunk-level temporal modeling on multiple datasets and different sequence-to-one tasks (e.g. age detection), (ii) explore other solutions for feature extraction such as CNN and DNN, and (iii) analyze insights derived from chunk-level attention weights to understand better the non-uniform externalization of emotion.

6. Acknowledgements

Study supported by NSF (CNS-1823166; IIS-1453781).
7. References


