Information Magnitude Based Dynamic Sub-sampling for Speech-to-text

Yuhao Zhang¹*, Chenghao Gao¹*, Kaiqi Kou¹, Chen Xu¹, Tong Xiao¹,²†, Jingbo Zhu¹,²

¹ School of Computer Science and Engineering, Northeastern University, Shenyang, China ² NiuTrans Research, Shenyang, China

{yuhao.zhang, koukq0907}@gmail.com, {gaochrishao, xuchenneu}@outlook.com, {xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Attention-based models have achieved new state-of-the-art in many tasks while the computational cost of these models increases drastically compared with previous methods. For most acoustic tasks, excessively long speech sequences exacerbate this problem and do not benefit a lot from the attention mechanism. We propose the information magnitude (IM) based dynamic stride convolution (IM-DSC) method. This method first calculates the IM according to the importance of each frame, then dynamically squeezes the redundant frames. We carry on experiments on speech translation and automatic speech recognition tasks. Our results show that we achieve 0.5 BLEU and 0.4 WER reduction on the Must-C En-De and En-Fr datasets with a 22% compression ratio. For the ASR task, we gain a 0.2 WER reduction with a 21% compression ratio on the Librispeech dataset.

Index Terms: attention-based model, speech-to-text, sequence compression

1. Introduction

The end-to-end model has gained increasing attention due to its impressive performance [1]. In comparison to hidden markov models [2] and recurrent neural networks (RNNs) [3], the attention-based model has a stronger ability to represent and extract information [4]. As a result, it has achieved state-of-the-art performance on both automatic speech recognition (ASR) and speech translation (ST) tasks [5, 6]. However, the attention-based model relies on nonlinear transfer and attention operations, which require a large number of parameters and multiplications operations. This leads to a drastic increase in the cost of training and brings difficulty to its application. In natural language processing (NLP) tasks, this problem has been noted, and numerous efficient models have been proposed [7, 8, 9, 10].

Unlike NLP tasks, acoustic processing tasks require speech waveforms to be converted to frames [11]. Consequently, the obtained sequence is significantly longer (e.g., dozens of times) than the corresponding text [12]. This increase in length causes a significantly higher cost of matrix computation, and the problem of computational latency is much serious [13]. Additionally, this also prevents the attention mechanism from effectively extracting data from noise [14].

To address these challenges, it is essential to reduce the length of frames, and some sub-sampling strategies have been proposed. Static sub-sampling methods use convolutional neural networks (CNNs) to aggregate features of adjacent frames [15, 16]. But it cannot get rid of noise and many silent frames are still present (see (a) of Figure 1). Dynamic sub-sampling methods were proposed to overcome these limitations. This method introduces an RNN module to identify the useful units, and only necessary frames are passed to the subsequent modules [17, 18]. Although this method achieves an impressive compression ratio, it cannot be parallelized, and errors in the RNN module can lead to performance degradation [12]. Moreover, the RNN-based strategy may capture meaningless frames and miss some essential frames [18] (see (b) of Figure 1).

We propose a parameter-free and efficient method called IM-DSC (Information Magnitude-based Dynamic Stride Convolution) that combines the advantages of the previous two methods. IM-DSC first generates an information magnitude (IM) for each frame using strategies such as GMM [19], SVM [20], etc. The IM metric represents the importance of each frame, i.e., a higher IM means that this frame contains more relevant information that should not be missed. Then, the convolution network uses the dynamic stride based on the IM to select whether to preserve useful information or compress noise (see (c) of Figure 1). The lightweight IM-DSC method significantly increases the density of information, namely, using minimal length to preserve all useful information.

We evaluated our proposed method on the Must-C dataset based on the strong baseline and found that IM-DSC achieved 0.4 to 0.5 BLEU improvements with a 22% compression ratio over the baseline model. Additionally, our method outperformed the Conformer-based baseline [21] on Librispeech, reducing WER by 0.21.

2. Related Work

Static stride CNN [12, 15] and max-pooling [16] are used to compress the frames along the time dimension in acoustic processing. But if we further compress the frame sequence, the proportion of redundant information (such as pauses, highly correlated adjacent frames, etc.) will not decrease, even worse, the important information will be over-sampled. Therefore, some researchers have proposed dynamic sub-sampling schemes [17, 18] based on RNNs. It can dynamically skip unimportant frames during recurrence and largely increases the reduction rate of sequence. But the computational latency is further exacerbated due to the recurrent nature of RNNs which can not be computed in parallel. If the downstream network is complex, this additional training strategy will increase the instability of training. Some researchers proposed progressive sub-sampling [22] to reduce the complexity of model computation, but this method requires embedding a sub-sampling module inner the conformer layer, which adding more parameters.

* Equal contribution
† Corresponding author

The code is available on https://github.com/GaoChrishao/GDS-Con.
3. Proposed Method: IM-DSC

The structure of IM-DSC is depicted in Figure 2. The acoustic features are sub-sampled by IM-DSC and then fed into the attention-based model. IM-DSC comprises the IM scoring module and the dynamic stride convolution layer (DSC). The IM scoring module assigns importance to each frame, and the DSC dynamically extracts features based on the calculated magnitude. We will describe these two modules in more detail in the following sections.

3.1. IM Scoring Module

The initial step is to obtain the IM, which represents the frame’s importance. A similar task is Voice Activity Detection [19, 25], which determines the presence of human speech in audio based on the frame features’ energy. Inspired by this, we first define the IM level in reference to the frame energy and then design the classification method to score it. Although there are many methods to calculate IM, we select Gaussian mixture model (GMM) to achieve this goal to avoid increasing computing costs and model parameters.

GMM is a combination of multiple Gaussian distributions and has been widely used in voice activity detection [19, 26, 27]. The GMM-based model can distinguish between noise and speech segments due to the higher discrimination between speech and non-speech regions. Previous work suggests that frames with valid information belong to a Gaussian distribution with a large mean and covariance, while non-speech frames belong to a distribution with a small mean and covariance [19]. This assumption is because non-speech frames are generally considered to be more stable than voice signals [28]. Therefore, the Gaussian distribution with a lower mean can represent more unimportant information. Given a speech frame vector \( x = (x_1, x_2, ..., x_d) \), \( d \) is dimension of speech vector, the probability density function of the entire mixture distribution can be expressed by the following formula:

\[
P(x) = \sum_{k=1}^{K} \pi_k \cdot N(x; \mu_k, \Sigma_k)
\]

where \( K \) is the number of distributions, and \( \pi_k, \mu_k, \) and \( \Sigma_k \) are the weight, mean vector, and covariance matrix of \( k \)-th distribution. Here, we use \( K = 2 \) to classify the IM metric of each frame. To get the maximum likelihood of \( P(x) \), we apply the expectation-maximization [29] algorithm to find appropriate \( \pi_k, \mu_k, \) and \( \Sigma_k \) in the first few training steps.

Then, the acoustic frames \((x_1, x_2, ..., x_n)\) are fed to the IM scoring module to generate the IM sequence \( m \), which could be described using the following formula:

\[
m_i = \arg \max_k(p(x_i|\pi_k, \Sigma_k, \mu_k))
\]

Here, the \( \arg \max() \) function chooses the \( k \) that maximizes the probability of the components \( p() \) of the above Eq.1. For example, an IM sequence may look like \((1, 2, 2, 1, 2)\), which indicates that \((x_1, x_4)\) belong to meaningless frames, and the rest can be classified as essential frames.

3.2. Dynamic Stride Convolution

Previous sub-sampling work usually uses CNNs with a static stride. For an input with \( n \) frames, the extracted length \( n_{static} \) more important information. Given a speech frame vector \( x = (x_1, x_2, ..., x_d) \), \( d \) is dimension of speech vector, the probability density function of the entire mixture distribution can be expressed by the following formula:

\[
P(x) = \sum_{k=1}^{K} \pi_k \cdot N(x; \mu_k, \Sigma_k)
\]

where \( K \) is the number of distributions, and \( \pi_k, \mu_k, \) and \( \Sigma_k \) are the weight, mean vector, and covariance matrix of \( k \)-th distribution. Here, we use \( K = 2 \) to classify the IM metric of each frame. To get the maximum likelihood of \( P(x) \), we apply the expectation-maximization [29] algorithm to find appropriate \( \pi_k, \mu_k, \) and \( \Sigma_k \) in the first few training steps.

Then, the acoustic frames \((x_1, x_2, ..., x_n)\) are fed to the IM scoring module to generate the IM sequence \( m \), which could be described using the following formula:

\[
m_i = \arg \max_k(p(x_i|\pi_k, \Sigma_k, \mu_k))
\]

Here, the \( \arg \max() \) function chooses the \( k \) that maximizes the probability of the components \( p() \) of the above Eq.1. For example, an IM sequence may look like \((1, 2, 2, 1, 2)\), which indicates that \((x_1, x_4)\) belong to meaningless frames, and the rest can be classified as essential frames.

3.2. Dynamic Stride Convolution

Previous sub-sampling work usually uses CNNs with a static stride. For an input with \( n \) frames, the extracted length \( n_{static} \) more important information. Given a speech frame vector \( x = (x_1, x_2, ..., x_d) \), \( d \) is dimension of speech vector, the probability density function of the entire mixture distribution can be expressed by the following formula:

\[
P(x) = \sum_{k=1}^{K} \pi_k \cdot N(x; \mu_k, \Sigma_k)
\]

where \( K \) is the number of distributions, and \( \pi_k, \mu_k, \) and \( \Sigma_k \) are the weight, mean vector, and covariance matrix of \( k \)-th distribution. Here, we use \( K = 2 \) to classify the IM metric of each frame. To get the maximum likelihood of \( P(x) \), we apply the expectation-maximization [29] algorithm to find appropriate \( \pi_k, \mu_k, \) and \( \Sigma_k \) in the first few training steps.

Then, the acoustic frames \((x_1, x_2, ..., x_n)\) are fed to the IM scoring module to generate the IM sequence \( m \), which could be described using the following formula:

\[
m_i = \arg \max_k(p(x_i|\pi_k, \Sigma_k, \mu_k))
\]

Here, the \( \arg \max() \) function chooses the \( k \) that maximizes the probability of the components \( p() \) of the above Eq.1. For example, an IM sequence may look like \((1, 2, 2, 1, 2)\), which indicates that \((x_1, x_4)\) belong to meaningless frames, and the rest can be classified as essential frames.

3.2. Dynamic Stride Convolution

Previous sub-sampling work usually uses CNNs with a static stride. For an input with \( n \) frames, the extracted length \( n_{static} \) more important information. Given a speech frame vector \( x = (x_1, x_2, ..., x_d) \), \( d \) is dimension of speech vector, the probability density function of the entire mixture distribution can be expressed by the following formula:

\[
P(x) = \sum_{k=1}^{K} \pi_k \cdot N(x; \mu_k, \Sigma_k)
\]

where \( K \) is the number of distributions, and \( \pi_k, \mu_k, \) and \( \Sigma_k \) are the weight, mean vector, and covariance matrix of \( k \)-th distribution. Here, we use \( K = 2 \) to classify the IM metric of each frame. To get the maximum likelihood of \( P(x) \), we apply the expectation-maximization [29] algorithm to find appropriate \( \pi_k, \mu_k, \) and \( \Sigma_k \) in the first few training steps.

Then, the acoustic frames \((x_1, x_2, ..., x_n)\) are fed to the IM scoring module to generate the IM sequence \( m \), which could be described using the following formula:

\[
m_i = \arg \max_k(p(x_i|\pi_k, \Sigma_k, \mu_k))
\]

Here, the \( \arg \max() \) function chooses the \( k \) that maximizes the probability of the components \( p() \) of the above Eq.1. For example, an IM sequence may look like \((1, 2, 2, 1, 2)\), which indicates that \((x_1, x_4)\) belong to meaningless frames, and the rest can be classified as essential frames.
\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Model & dev-clean & dev-other & test-clean & test-other & R(\%) \\
\hline
Baseline_{2 \times 2} & 3.04 & 7.02 & 3.02 & 7.41 & 25 \\
Baseline_{4 \times 4} & 2.97 & 7.62 & 2.99 & 7.70 & 13 \\
T-DSC & 3.19 & 7.23 & 3.08 & 7.31 & 21 \\
IM-DSC & 2.88 & 7.10 & 2.85 & 7.17 & 21 \\
\hline
\end{tabular}
\caption{WER results on Librispeech (without language model scoring).}
\end{table}

is computed by the following formula:

\[ n_{\text{static}} = \frac{n - c}{s} + 1 \quad (3) \]

where \( c \) is the kernel size and \( s \) is the stride size. We design the dynamic stride convolution (DSC) layer which reduces the number of non-speech frames based on the calculated IM sequence \( m \). We define the dynamic stride set \( S = \{ s_1, s_2, \ldots, s_K \} \) where \( K \) is defined by the IM scoring module. We split the speech sequence \( (x_1, x_2, \ldots, x_n) \) into segments by a fixed window-size \( w \). \( w \) is the max value in \( S \) and this can avoid the possible loss if this slice uses the max stride. Considering the stationarity of the speech signal, we use the IM* to represent the IM of one segment and it can be decided by the IM with the most occurrences in this segment. If IM* equals to \( k^* \), we can use the corresponding \( s_{k^*} \) to extract the features of adjacent frames. If the IM* of this segment is low, then the stride will be larger to aggregate less information from this segment which is likely to be noise. The output length of the DSC layer \( n_{\text{DSC}} \) can be computed by the following formula:

\[ n_{\text{DSC}} = \sum_{k=1}^{K} l_k - c + 1 \quad (4) \]

where \( l_k \) denotes the total length of frames belonging to the \( k \)-th IM. We can achieve a greater compression ratio by adjusting each \( s_k \) larger than \( s \) and still preserve the useful information.

4. Experiments

4.1. Preprocessing

We evaluated the proposed IM-DSC on the Librispeech data [31] for the ASR task and the MuST-C English-German and English-French datasets for the ST task [32]. We followed the recipe of Fairseq-S2T [23] for preprocessing all datasets. We used a 25ms window with a shift of 10ms to extract 80-dimensional log-mel filterbank data and applied SpecAugment [33] to augment the speech data. We also used Byte-Pair Encoding [34] subword segmentation with a size of 10,000 to build the shared vocabulary for every dataset.

4.2. Model Settings

For the ST tasks, we used the Transformer architecture as the baseline model, which consisted of 12 layers of encoder and 6 layers of decoder. All layers had a hidden size of 256, 4 attention heads, and 2048 feed-forward size. We used CTC [35] loss with a weight of 0.3 to assist training. The sub-sampling layer used two stacked CNNs with a stride of 2 and a conv size of 5 to compress the input acoustic features.

For the ASR task, we used the mainstream architecture Conformer [21] as the baseline model. It included 12 Conformer layers of encoder and 6 Transformer layers of decoder. The other hyperparameters were similar to ST task.

In our IM-DSC, we began by sampling 2,000 sentences from the training set to identify suitable GMM parameters for each task. We then froze these GMM parameters during the subsequent training process. To achieve a compression ratio greater than 4, we employed two CNNs with a kernel size of 5, stacked together. The CNN layers’ stride had a dynamic stride set of \( S = \{2, 4\} \).

During inference, we averaged the last 10 model parameter checkpoints and used a beam size of 5 for improved decoding. We report all experiments using SacreBLEU [36] for ST tasks and Word Error Rate (WER) for the ASR task.

4.3. Results on ASR and ST

Table 1 displays the results for the MuST-C En-De and En-Fr tasks. Compared to the baseline with 2 \times 2 times sub-sampling, the model with 4 \times 4 times sub-sampling experiences significant performance degradation. However, if we apply the IM-DSC method, the model outperforms the baseline model with 0.4 BLEU and 0.5 BLEU points at a higher compression ratio. This indicates that simply compressing the length will lead to loss of useful information, but with the guidance of IM, our IM-DSC can accurately drop redundant information. Moreover, the attention operation can more easily extract useful information due to the reduced noise in the sequence. Thus, IM-DSC outperforms the baseline and is faster.

A similar trend is observed in the ASR task, as shown in Table 2, which confirms the findings from the ST task. The robust Conformer only decreases much on the test-other set when the compression ratio becomes higher. However, our IM-DSC remains effective with this strong and robust model, which demonstrates the good generalization of our method.

We use the energy threshold to generate the IM sequence, which is called T-DSC. Specifically, we define that frames with acoustic features below the threshold have a low IM. To achieve the same compression ratio as IM-DSC, we set the threshold to -0.2. As shown in Table 2 and 1, T-DSC underperforms IM-DSC, indicating that GMM-based IM is accurate and essential.

5. Analysis

5.1. Effect of IM

We sampled a speech from the training dataset as a case study and illustrated its IM tagging result based on the GMM in Figure 3. The frames that align with the word have higher energy, thus, using the classification strategy to tag the IM is a natural choice. The tagging result displays that all redundant and important frames correspond precisely to their respective IM. This demonstrates that our GMM can accurately classify the speech
frames and guide the sub-sampling operation. We further investigated the behavior of model after directly removing frames with low IM. Table 3 shows that regardless of whether we remove non-speech frames before or after static sub-sampling, performance degrades. We observed that the performance loss of baseline\(_{\text{c}}\) which preserves some non-speech features by sub-sampling CNNs, is significantly smaller than that of baseline\(_{\text{a}}\). This indicates that some frames with low IM contain necessary pause and boundary information. Integrating this essential information using our DSC rather than dropping or skipping it is a more reasonable approach.

To test the robustness of DSC, we replaced 10\% of the IM sequence’s area with the wrong tagging, resulting in IM-DSC-d. As shown in Table 3, IM-DSC-d can still preserve essential information, and performance did not suffer significant degeneration on all tasks.

5.2. Impact on Attention Mechanism

Figure 4 shows the change in attention weight after using IM-DSC. The pause between words “a” and “trick” is compressed by IM-DSC, resulting in a reduction in frame size. Further, due to the reduction of noise, the attention distribution tends to focus on useful positions, such as “trick” and “question”. This explains why IM-DSC can improve both speed and accuracy.

Attention operation is difficult to handle long sequences. Thus if we sub-sample the speech that has a long length, the model should exhibit better improvement than the short ones. We tested this hypothesis on the speech translation task, as cross-lingual tasks rely heavily on the attention mechanism. We split the MuST-C En-De dataset into two sets according to the compression ratio calculated by IM-DSC. Table 4 shows that the IM-DSC achieves slight improvement in short sequences and a greater improvement in long sentences, which confirms our proposal. Noteworthily, the baseline method with 8x compression ratio leads to great performance degradation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg length</th>
<th>R(%)</th>
<th>BLEU</th>
<th>Δ BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline(_{2 \times 2})</td>
<td>688</td>
<td>25</td>
<td>23.17</td>
<td>-</td>
</tr>
<tr>
<td>Baseline(_{4 \times 2})</td>
<td>688</td>
<td>13</td>
<td>22.46</td>
<td>0.71↓</td>
</tr>
<tr>
<td>IM-DSC</td>
<td>668</td>
<td>21</td>
<td>23.54</td>
<td>0.37↑</td>
</tr>
<tr>
<td>Baseline(_{2 \times 2})</td>
<td>422</td>
<td>25</td>
<td>22.39</td>
<td>-</td>
</tr>
<tr>
<td>Baseline(_{4 \times 2})</td>
<td>422</td>
<td>13</td>
<td>21.21</td>
<td>1.18↓</td>
</tr>
<tr>
<td>IM-DSC</td>
<td>422</td>
<td>23</td>
<td>22.58</td>
<td>0.19↑</td>
</tr>
</tbody>
</table>

Table 4: Comparison of performances on two ST test sets with different lengths.

5.3. Training speed

To test the training speed fairly, we controlled the batch size and training steps on the Librispeech clean-100 set. The results of speed and convergence are shown in Table 5. Compared with the baseline, we found that IM-DSC achieved a stable 7\% acceleration without loss. This proves improving the down-sampling rate is an effective way to accelerate. Although the model needs to compute the IM metric for each frame first, the overall training speed is not significantly affected. This demonstrates that the GMM is an efficient method. Regarding the convergence speed, although this method showed little advantage in the early stages of training, the IM-DSC’s speed can be much faster than the baseline’s in later stages due to the noise reduction. This phenomenon also confirms our motivation.

6. Conclusion

We propose the IM-DSC, a novel sub-sampling method for acoustic tasks to dynamically compresses speech features. This method utilizes IM scoring module to obtain information magnitude, which guides the dynamic stride CNN to retain valid information and compress useless information. Experiments show that our method achieves superior performance and higher compression rate. Our analysis shows that the IM scoring module can accurately identify redundant information, and the dynamic stride CNN is robust when compressing speech features.

7. Acknowledgements

This work was supported by part in the National Science Foundation of China (No.62276056), the National Key R&D Program of China, the China HTRD Center Project (No.2020AAA0107904), the Natural Science Foundation of Liaoning Province of China (2022-KF-16-01), the Yunnan Provincial Major Science and Technology Special Plan Projects (No.202103AA080015), the Fundamental Research Funds for the Central Universities (Nos. N2216016, N2216001, and N2216002), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).
8. References


