Semantic VAD: Low-Latency Voice Activity Detection for Speech Interaction

Mohan Shi\textsuperscript{1}, Yuchun Shu\textsuperscript{2}, Lingyun Zuo\textsuperscript{3}, Qian Chen\textsuperscript{3}, Shiliang Zhang\textsuperscript{3}, Jie Zhang\textsuperscript{1}, Li-Rong Dai\textsuperscript{1}

\textsuperscript{1}NERC-SLIP, University of Science and Technology of China (USTC), China
\textsuperscript{2}Tianjin University, China
\textsuperscript{3}Speech Lab of DAMO Academy, Alibaba Group, China

Abstract

For speech interaction, voice activity detection (VAD) is often used as a front-end. However, traditional VAD algorithms usually need to wait for a continuous tail silence to reach a preset maximum duration before segmentation, resulting in a large latency that affects user experience. In this paper, we propose a novel semantic VAD for low-latency segmentation. Different from existing methods, a frame-level punctuation prediction task is added to the semantic VAD, and the artificial endpoint is included in the classification category in addition to the often-used speech presence and absence. To enhance the semantic information of the model, we also incorporate an automatic speech recognition (ASR) related semantic loss. Evaluations on an internal dataset show that the proposed method can reduce the average latency by 53.3\% without significant deterioration of character error rate in the back-end ASR compared to the traditional VAD approach.

Index Terms: Voice activity detection, semantic loss, low latency, speech interaction, character error rate.

1. Introduction

Speech interaction \cite{1} is one of the most representative applications of artificial intelligence, where long audio recordings need to be cut into short audio segments for downstream tasks, e.g., automatic speech recognition (ASR) \cite{2}, speech translation \cite{3}, speech emotion recognition \cite{4}, etc. For this purpose, a front-end voice activity detection (VAD) \cite{5–7} is often built to distinguish between speech and non-speech periods, as the former is more related to the interaction goals.

Over the last few decades, many VAD approaches have been proposed. Early VAD methods classify speech and non-speech regions based on energy \cite{8–12}, zero-crossing rate \cite{13}, and periodicity measure \cite{14}. More recently, with the advance in deep learning, the application of deep neural networks (DNNs) has become dominant in this field \cite{15–18}, where most aim to improve the classification capability. Also, there are some works on improving noise robustness, e.g., \cite{19–21}.

Directly applying these traditional VAD approaches may cause a new problem in realistic speech interaction scenarios. Specifically, traditional VAD algorithms usually need to wait for a continuous tail silence to reach a preset maximum tail silence duration (e.g., 700ms) before deciding whether to perform tail segmentation, e.g., see the example in Figure 1, which results in a relatively high tail segmentation latency and seriously affects the user experience \cite{22,23}. In order to address the high-latency issue of tail segmentation, we consider a novel semantic VAD approach in this work.

Firstly, as traditional VAD models only distinguish speech or silence and neglect whether each part of silence is a complete semantic breakpoint, they have to wait for a long continuous tail silence (e.g., 700ms) before making a tail segmentation judgment to avoid splitting the whole sentences incorrectly. It follows that if a semantic breakpoint is contained within the silence segment, the tail segmentation decision can then be simply made by waiting for a relatively short tail silence. In order to obtain semantic breakpoint information, we thus consider adding a frame-level punctuation prediction task to the basic VAD model. In case an ending punctuation is detected (e.g., period, question mark), indicating that there is a full semantic breakpoint, it is reasonable to wait for a short tail silence (e.g., 300ms). In the case of detecting a non-ending punctuation (e.g., comma), the tail silence required for segmentation needs to be slightly longer than that of ending punctuation (e.g., 400ms). Only when punctuation cannot be predicted, the preset maximum tail silence of the traditional VAD (e.g., 700ms) will be used to determine the segmentation point.

Second, in order to further increase the proportion of early judgments and reduce the tail segmentation latency, we add predictions for artificially defined endpoints. Specifically, the endpoint label is introduced in addition to conventional binary speech-presence and speech-absence categories, forming a three-class classification task. The endpoint label is generated by a combination of punctuation and silence duration, similar to how the punctuation is utilized to determine the tail segmentation. The difference lies in that combining punctuation to determine the tail segmentation is implemented in the inference stage, while combining punctuation to generate endpoint labels artificially changes the training target of VAD.

Finally, since the prediction of punctuation and endpoint requires the model to be semantically strong, we introduce a semantic loss as an auxiliary task to enhance the semantic modeling ability. It should be noted that, different from existing joint VAD and ASR multi-task training methods \cite{24,25}, where VAD is cascaded with ASR so that the speech is processed as a whole, resulting in a relatively large parameter amount, while...
the proposed semantic loss is only used to assist model training and not present at inference time. As such, the independence and lightweight of our semantic VAD model can be guaranteed, meaning that the proposed VAD model can be separately applied or flexibly combined with other downstream tasks.

Evaluations on an internal dataset show that the proposed method can reduce the average tail segmentation latency by 53.3% without significant deterioration of character error rate in the back-end ASR compared to the traditional VAD approach. The rest of this paper is organized as follows. Section 2 presents the proposed semantic VAD. Section 3 shows the experimental setup, followed by results in Section 4. Finally, Section 5 concludes this work.

2. Semantic VAD

The schematic of semantic VAD is shown in Figure 2, which consists of the speech encoder, punctuation prediction, VAD classifier, and semantic task.

2.1. Speech encoder

To facilitate the subsequent migration to the streaming platform, we use contextual block conformer [26, 27] as our speech encoder, which is a block processing conformer structure designed for streaming ASR. In the speech encoder, a context embedding vector is introduced to convey contextual information between blocks. The acoustic features are passed through the speech encoder to generate a frame-level speech representation.

2.2. Punctuation prediction

In traditional VAD, there are no decisions on semantic breakpoints, so it is necessary to wait for a long tail silence before performing tail segmentation. In order to reduce the tail segmentation latency, it might be helpful to obtain semantic segmentation via punctuation prediction. This can be accomplished by directly incorporating the frame-level speech representation obtained by the speech encoder into the classifier to obtain the frame-level punctuation prediction. In this work, we consider three types of punctuations:

\[
P = \begin{cases} 
\text{Not punctuation} \\ 
\text{Ending punctuation (E-punc)} \\ 
\text{Non-ending punctuation (NE-punc)} 
\end{cases} \tag{1}
\]

where the ending punctuation means complete semantic breakpoints (e.g., periods, question marks) and the non-ending punctuation implies incomplete semantic breakpoints (e.g., comma).

2.3. VAD with endpoint

Conventionally, each speech frame is only simply labeled with speech-presence or speech-absence. In order to further improve the proportion of tail segmentation points judged in advance and reduce the tail segmentation latency, the artificially defined endpoint label is introduced as the third category, resulting in a three-class classification problem:

\[
S \in \{ \text{Speech, Silence, Endpoint} \}, \tag{2}
\]

where the endpoint is the newly defined label in the training set. On the basis of obtaining the labels of speech and silence, in case a sentence ends with an E-punc, the frame after a continuous silence until the next speech begins will be set with an endpoint; in case of a NE-punc, the duration of the tail silence is then set to be \( t_{\text{NE}} \). Similarly to punctuation prediction, the output of the speech encoder has to pass through a classification layer to obtain the frame-level VAD decision.

2.4. Semantic task

Since both punctuation classification and the prediction of the artificially specified endpoint frames in the considered VAD, the representation obtained by the speech encoder should contain strong semantic information. Therefore, it is necessary to introduce a semantic loss as an auxiliary task to preserve the semantic information in speech representation. In this work, we use the ASR loss as the auxiliary semantic loss. Specifically, the speech encoder is connected to the decoder for pre-training and VAD label, respectively, while \( \hat{p}, \hat{y}, \) and \( \hat{s} \) represent the true punctuation, transcription, and VAD label, respectively, while \( \hat{p}, \hat{y}, \) and \( \hat{s} \) represent the corresponding hypothesis. Notice that unlike previous multi-task works, ASR is only used here as an auxiliary task to enrich the semantic information during training, and does not appear during inference. This guarantees the lightweight and independence of the front-end VAD module at the inference phase.

2.5. Inference

For traditional VAD methods, after the tail silence reaches \( t_{\text{Max}} \) continuously, it will be assigned a break point, and segmentation is performed at the beginning of the tail silence. This leads...
Table 2: Performance comparison of traditional and semantic VAD methods on the three test sets in terms of the tail segmentation latency (ms), CER (%) for back-end ASR and DCF (%) for front-end segmentation.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Average</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Latency</td>
<td>CER</td>
<td>DCF</td>
<td>Latency</td>
</tr>
<tr>
<td>Traditional VAD</td>
<td>700</td>
<td>16.75</td>
<td>10.78</td>
<td>700</td>
</tr>
<tr>
<td></td>
<td>600</td>
<td>16.80</td>
<td>10.78</td>
<td>600</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>16.98</td>
<td>10.70</td>
<td>500</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>17.42</td>
<td>10.71</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>330</td>
<td>17.94</td>
<td>10.70</td>
<td>330</td>
</tr>
</tbody>
</table>

Semantic VAD | 326.92  | 16.88 | 9.60  | 326.45  | 19.91 | 11.52 | 338.99  | 27.84 | 9.94  | 326.32  | 7.04  | 7.10  |

Table 1: The label distributions (%) in the training set.

<table>
<thead>
<tr>
<th>Punctuation labels</th>
<th>VAD labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not punc</td>
<td>Speech</td>
</tr>
<tr>
<td>E-punc</td>
<td>Silence</td>
</tr>
<tr>
<td>NE-punc</td>
<td>Endpoint</td>
</tr>
<tr>
<td>73.79</td>
<td>58.81</td>
</tr>
<tr>
<td>22.05</td>
<td>18.80</td>
</tr>
<tr>
<td>4.16</td>
<td>22.69</td>
</tr>
</tbody>
</table>

3. Experimental Setup

3.1. Dataset

In this work, we use 325 hours of internal Mandarin speech data, obtained from various websites, such as YouTube, as training and validation sets. We use an internal Mandarin long speech test set of three different scenarios for the final evaluation, including a meeting (Test1, for 3.3 hours), a recording pen (Test2, for 1 hour), and a live broadcast (Test3, for 2.1 hours). Punctuation labels per frame are obtained by forcing alignment. Table 1 shows the distribution of various labels in the training set.

3.2. Evaluation metrics

Latency: Since the focus of this work is on the reduction of the tail latency of VAD segmentation during speech interaction, the latency time is taken as one of the evaluation metrics. Specifically, the latency is measured by the average continuous tail silence duration during VAD segmentation. Since the same VAD model is used in experiments, the latency of the model itself is ignored for convenience.

Character error rate (CER): Because the downstream task of the back-end ASR is the recognition of the audio, the CER is used as the evaluation metric. A higher CER means a less accurate VAD segmentation.

3.3. Model configuration

In this work, we use the 80-dimensional log-mel filterbank (Fbank) as the feature input. The window size is 25 ms, and the hop size is 16 (i.e., 160 ms). The convolution kernel size is set to be 15, the block size is 64 (corresponding to 640 ms), and the hop size is 16 (i.e., 160 ms). In each block, the first 160 ms is the future information. For the auxiliary ASR task, a mixture of 4-layer transformer decoder and CTC [29] is used for training. The decoder configuration keeps the same as the speech encoder with a CTC loss weight of 0.3. The parameter amount of the speech encoder is around 5.98M. We first pre-train the ASR task for 50 epochs, and then jointly train the ASR and VAD tasks for 50 epochs, where both λ and μ are set to be 2.0. We use the ESPnet toolkit [30] for our experiments.

4. Experimental Results

First of all, we show in Table 2 the comparison of the performance of traditional and semantic VAD methods on the three test sets in terms of the tail segmentation latency, back-end ASR and front-end segmentation. For a fair comparison, the same speech encoder structure as the semantic VAD is used in the traditional VAD, but the CTC loss in the training stage is ignored. That is, only the binary classification is performed by the traditional VAD approach to distinguish speech from silence, and the semantic loss is not considered during training. It can be seen...
Table 3: Performance comparison of the performance of semantic VAD under different strategies on the three test sets.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Average</th>
<th>Test1</th>
<th>Test2</th>
<th>Test3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Latency</td>
<td>CER</td>
<td>DCF</td>
<td>Latency</td>
</tr>
<tr>
<td>Semantic VAD</td>
<td>362.95</td>
<td>16.92</td>
<td>10.91</td>
<td>355.29</td>
</tr>
<tr>
<td>+Endpoint</td>
<td>326.92</td>
<td>16.88</td>
<td>9.60</td>
<td>326.45</td>
</tr>
</tbody>
</table>

that in traditional VAD, the performance of back-end ASR becomes worse (with an average CER increase from 16.75% to 17.94%) as the preset tail segmentation latency decreases from 700 ms to 330 ms. The proposed semantic VAD approach can significantly reduce the average tail segmentation latency (from 700 ms to 326.92 ms) without a serious sacrifice in the back-end ASR performance (with an average CER increase from 16.75% to 16.88%). More importantly, the proposed method can improve the front-end segmentation performance in DCF from 10.78% to 9.60% due to more semantic segmentation. It is worth mentioning that in case the tail segmentation latency of traditional VAD is set to be 330 ms, it becomes equivalent to the proposed semantic VAD, while the corresponding performance of back-end ASR is much worse than that of semantic VAD. These clearly show that the proposed method can not only reduce the latency and improve the segmentation accuracy, but also preserve the back-end ASR performance.

Table 3 presents the comparison of the proposed semantic VAD with different strategies on the three test sets in terms of the tail segmentation latency, back-end ASR and front-end segmentation. It is clear that adding the semantic loss as an auxiliary task in training leads to better performance in all three metrics (e.g., latency from 362.95ms to 344.63ms, CER from 16.75% to 16.88%, DCF from 10.78% to 9.60% on average). This indicates that with the assistance of semantic loss, the proportion of segmentation through punctuation and endpoint becomes larger, and the semantic of the segmented sentence is more complete, which is informative to the back-end ASR module. Involving endpoint judgment in inference leads to a further reduction in tail segmentation latency and DCF, but slightly degrades the back-end ASR performance. This is due to the fact that the endpoint is a kind of artificially defined label, which somehow increases the error in inference.

Finally, we conduct a visual analysis of the proposed semantic VAD. In Figure 4, we compare the tail judgment points of traditional VAD and semantic VAD for an audio segment of Test 1. The upward and downward stems denote the judgment points of the traditional and proposed VAD methods, and the red, blue, and green stems represent the judgments of E-punc, NE-punc, and Endpoint, respectively. It is clear that the judgment points of the semantic VAD are all before those of the traditional VAD. This verifies the efficacy of the proposed method in reducing latency. Figure 5 illustrates the proportion of various ways of obtaining tail judgment points for the semantic VAD on all test sets. It can be clearly seen that the maximum silence judgment method (i.e., traditional VAD) accounts for only 5.41%, which indicates that our semantic VAD method can make most tail segmentation points be judged in advance. As the endpoint accounts for a relatively small proportion (14.42%), the artificially defined label is semantically abstract, and it is thus necessary to make a pre-judgment through punctuation prediction (e.g., with high proportions).

5. Conclusion

In this paper, we proposed a novelty semantic VAD method, which can synthetically determine whether tail segmentation should be performed by introducing the punctuation prediction, artificially defined endpoints, and auxiliary semantic task. Experiments on the internal dataset show that the proposed semantic approach significantly reduces the tail segmentation latency without an obvious sacrifice in the back-end ASR performance compared to the traditional VAD approach. In the future, we will investigate the application to other semantic-dependent speech tasks (e.g., speech translation, dialogue understanding) as well as the generalization to multi-task cases.
6. References


