CONTINUOUS SIGN LANGUAGE RECOGNITION VIA REINFORCEMENT LEARNING

Zhihao Zhang, Junfu Pu, Liansheng Zhuang, Wengang Zhou, Houqiang Li

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System
EEIS Department, University of Science and Technology of China
{zzh3, pjh}@mail.ustc.edu.cn, {lszhuang, zhwg, lihq}@ustc.edu.cn

ABSTRACT

In this paper, we propose an approach to apply the Transformer with reinforcement learning (RL) for continuous sign language recognition (CSLR) task. The Transformer has an encoder-decoder structure, where the encoder network encodes the sign video into the context vector representation, while the decoder network generates the target sentence word by word based on the context vector. To avoid the intrinsic defects of supervised learning (SL) in our task, e.g., the exposure bias and non-differentiable task metrics issues, we propose to train the Transformer directly on non-differentiable metrics, i.e., word error rate (WER), through RL. Moreover, a policy gradient algorithm with baseline, which we call Self-critic REINFORCE, is employed to reduce variance while training. Experimental results on RWTH-PHOENIX-Weather benchmark verify the effectiveness of our method and demonstrate that our method achieves the comparable performance.

Index Terms—sign language recognition, reinforcement learning, self-critic

1. INTRODUCTION

Millions of hearing-impaired people routinely use some variants of sign languages to communicate, however, it’s difficult to understand sign language for the hearing society. As a result, there is a huge communication disorder between the deaf-mute and the hearing people, which makes the automatic translation of sign language meaningful and important.

Continuous sign language recognition (CSLR) aims at translating sign videos into text sentences. Though significant progress [1, 2, 3, 4] has been made, CSLR is still a very challenging task. It requires a fine-grained understanding of gestures, hand motions or even facial expressions in a video. Meanwhile, there exist semantic gaps between videos and sentences, as well as the difficulty of frame or word level alignment. To solve these challenges, we propose our CSLR model, as shown in Fig. 1. First, we adopt a 3D convolutional neural network to extract visual features from sign videos. Recently, residual network (ResNet) [5] and 3D convolutional neural network (3D CNN [6, 7, 8]) have shown outstanding performance in image and video representation, respectively. Inspired by the superiorities of ResNet and 3D CNN, we employ a combined 3D residual convolutional neural network (3D-ResNet) for feature extraction following CNN-DCN [9]. Second, we utilize a powerful neural machine translation (NMT) model to translate sign videos into text sentences. Recently, the Transformer [10], the first sequence transduction model based entirely on attention, achieves state-of-the-art performance on the English-German and English-French translation tasks. Considering the similarity between NMT and CSLR, we adopt the Transformer to bridge the semantic gap between sign videos and text sentences.

However, the Transformer [10] (or LSTM [11], etc.) for sequence transduction is typically trained to maximize the likelihood of the next ground-truth word given the previous ground-truth word using error back-propagation. This approach suffers a mismatch between training and testing since at test-time the model uses the previously generated words from the model distribution to predict the next word. This exposure bias [12] results in error accumulation during generation at the test time, since the model has never been exposed to its own predictions. Besides, there exist deviation between optimization objectives, i.e., the cross-entropy loss, during training and the non-differentiable evaluation metrics during testing. Recently, it has been shown that both the exposure bias and non-differentiable task metrics issues can be addressed through reinforcement learning (RL). Motivated
by these works [13, 14, 15], we employ REINFORCE [16] to train our CSLR model. Moreover, we append a baseline to REINFORCE to form a self-critic architecture because of the high variance of REINFORCE.

In summary, our major contributions are listed as follows:

- We propose a novel framework based on 3D-ResNet and the Transformer for continuous sign language recognition (CSLR). To the best of our knowledge, we are the first to deploy the Transformer for sequence learning in CSLR.

- We introduce an RL-based optimization strategy for our CSLR model. Experiments on the RWTH-PHOENIX-Weather demonstrate the effectiveness of our approach.

2. RELATED WORK

In this section, we briefly review some continuous sign language recognition (CSLR) methods, and compactly introduce some sequence generation tasks which are closely related to our work.

CNN-LSTM based methods [17, 18] are very popular for continuous sign language recognition (CSLR). Recently, some works such as [19, 20] have employed a CNN-LSTM network with connectionist temporal classification (CTC) [21] for CSLR, since CSLR task lacks supervision on accurate temporal segmentation for sign words. In addition, there are some approaches for CSLR which are based on other sequential models. Re-Sign [22] embeds a hidden markov model (HMM) into a deep recurrent CNN-BLSTM network with an iterative re-alignment approach for CSLR. CNN-DCN [9] proposes a deep neural architecture composed by 3D-ResNet and dilated convolutional network [23] with CTC loss for CSLR. Similarly, we employ the Transformer [10] as the sequential model instead of LSTM to solve the CSLR task.

Neural machine translation (NMT) is a typical sequence learning task and has drawn much attention. Recently, the Transformer [10] has achieved the state-of-the-art results on both WMT2014 English-German and English-French translation tasks. Moreover, BR-CSGAN [24] proposes an approach for applying GANs to NMT with the Transformer through policy gradient methods. Actually, reinforcement learning (RL) algorithms are widely used among sequence learning tasks. MIXER [12] adopts the REINFORCE [16] algorithm for text generation applications. Since REINFORCE suffers from high variance, it requires a proper baseline. Therefore, Bahdanau et al. [14] train another critic network to predict the value of an output token. SCST [13] utilizes the output of its own test-time inference algorithm to normalize the rewards it experiences. Inspired by these works, we utilize REINFORCE with a baseline to train our model.

3. OUR METHODS

In this section, we first propose a novel architecture based on the Transformer for continuous sign language recognition (CSLR). Then, we introduce our Self-critic REINFORCE for network training in our CSLR model.

3.1. Model Architecture

Our CSLR model consists of two components: 3D-ResNet for extracting video clip feature, and the Transformer which translates the visual feature sequences into sentences.

3D-ResNet. It is a great challenge to extract semantic information of sign language, which is contained in those elements of gestures, hand motions or even facial expressions in videos, for CSLR. Fortunately, 3D CNN has shown strong capability for video representation based on spatio-temporal information, since it considers the sequential relationship by temporal connections across frames. Following CNN-DCN [9], we adopt the 18-layers 3D-ResNet, which only replaces the 2D convolutional filters with 3D convolutional filters, for feature extraction. Furthermore, the training method for 3D-ResNet is the same as that introduced in CNN-DCN [9] as well.

The Transformer. The Transformer [10], as shown in Fig. 2, with an encoder-decoder architecture, has shown strong capability in neural machine translation (NMT). The encoder of the Transformer is composed of a stack of N identical layers. Each layer consists of a multi-head self-attention and a simple position-wise fully connected feed-forward network. The decoder is also composed of a stack of N identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. In the CSLR task, the Transformer regards the sign videos as source
language and translates the source language to the target language (i.e., text sentences). The input of the encoder is the video clip feature sequence extracted by 3D-ResNet. The input of the decoder is the learned embedding of a sentence or the beginning token of a sentence, while the decoder outputs the predicted next-token probabilities.

3.2. Self-critic REINFORCE

CSLR as a RL problem. Generally, the Transformer is trained using the cross-entropy loss. However, there exists deviation between the cross-entropy loss and the non-differentiable evaluation metrics, i.e., WER. To directly optimize the WER metric, we cast our models in the reinforcement reanning (RL) terminology. The Transformer, which can be viewed as an “agent”, defines a policy \( p_\theta \). The “agent” consistently produces the “action”, i.e., the prediction of the next token. We define the immediate reward \( r = 0 \) until the end-of-sequence (EOS) token generates. And “1 - WER” is received as the terminal reward denoted by \( R \). To minimize the negative expected cumulative reward with the discount rate \( \lambda = 1 \), we formulate the goal as follows,

\[
L(\theta) = -\mathbb{E}_{\omega^s \sim p_\theta} [R(\omega^s)],
\]

where \( \omega^s = (\omega^s_1, \cdots, \omega^s_T) \) and \( \omega^s_t \) is the word sampled from the model at the time step \( t \).

**REINFORCE with baseline.** According to [25], we compute the gradient \( \nabla L(\theta) \),

\[
\nabla L(\theta) = -\mathbb{E}_{\omega^s \sim p_\theta} [\nabla_{\theta} \log p_\theta(\omega^s)].
\]

We use samples of the expectation to instantiate our generic stochastic gradient ascent algorithm,

\[
\nabla L(\theta) \approx -R(\omega^s) \nabla_{\theta} \log p_\theta(\omega^s).
\]

This algorithm is called REINFORCE [16]. In addition, the policy gradient given by REINFORCE can be generalized to include a comparison of the action value \( R(\omega^s) \) to an arbitrary baseline \( b \) as long as it does not depend on the “action” \( \omega^s \),

\[
\nabla L(\theta) = -\mathbb{E}_{\omega^s \sim p_\theta} [(R(\omega^s) - b) \nabla_{\theta} \log p_\theta(\omega^s)].
\]

For each training case, we again approximate the expected gradient with a single sample \( \omega^s \sim p_\theta \),

\[
\nabla L(\theta) \approx -(R(\omega^s) - b) \nabla_{\theta} \log p_\theta(\omega^s).
\]

Self-Critic REINFORCE. The central idea of the Self-critic REINFORCE is to take the reward, which is obtained by the Transformer under the inference algorithm used at the test time, as the baseline for the REINFORCE. As shown in Fig. 3, \( R(\omega^s) \) is the reward which represents the sentence (i.e., \( (\omega^s_1, \cdots, \omega^s_T) \)) generated through sampling based on its probabilities. Similarly, \( \hat{R}(\hat{\omega}) \) is the reward which evaluates the sentence (i.e., \( (\hat{\omega}_1, \cdots, \hat{\omega}_T) \)) obtained by the Transformer under the inference algorithm used at test time, i.e.,

\[
\hat{\omega}_t = \arg \max_{\omega_t} p(\omega_t).
\]

For each training case, the Self-critic REINFORCE gives:

\[
\nabla L(\theta) \approx -(R(\omega^s) - \hat{R}(\hat{\omega})) \nabla_{\theta} \log p_\theta(\omega^s).
\]

Self-critic REINFORCE inherits all the advantage of REINFORCE, as it not only directly optimizes the true, sequence-level evaluation metric but also avoids the usual scenario of having to learn a context-dependent estimate of expected future rewards as a baseline. Since the Self-critic REINFORCE baseline is based on the test-time estimation under the current model, Self-critic REINFORCE is forced to improve the performance of the model under the inference algorithm used at the test time. Besides, Self-critic REINFORCE avoids all the inherent training difficulties associated with actor-critic methods, where a second “critic” network must be trained to estimate value functions, and the actor must be trained on estimated value functions rather than actual rewards.

4. EXPERIMENTS

In this section, we first introduce the experiment setup. Besides, we discuss the comparison results as well as the ablation study.
Table 1. Summary of RWTH-PHOENIX-Weather dataset

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
<th>Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Sentences</td>
<td>5672</td>
<td>629</td>
<td>540</td>
</tr>
<tr>
<td>#Vocabulary</td>
<td>1231</td>
<td>497</td>
<td>461</td>
</tr>
<tr>
<td>#Words</td>
<td>65227</td>
<td>6530</td>
<td>5564</td>
</tr>
</tbody>
</table>

4.1. Experiment Setup

Dataset. We conduct our experiments on the German sign language dataset, i.e., RWTH-PHOENIX-Weather [26]. The dataset contains 7K weather forecast sentences from 9 signers. All videos are of 25 frames per second (FPS) and at resolution of 210 × 260. Following [26], 5,672 instances are used for training, 540 for validation, and 629 for testing. The statistic details of this dataset are available in Table 1.

Evaluation Metrics. Predicted sentence may suffer from errors including word substitution, insertion and deletion error. Following [18, 27, 28, 29], we measure the performance with word error rate (WER),

\[ WER = \frac{S + I + D}{N} \times 100\%, \]  

where \( S \), \( I \) and \( D \) denote the minimum number of substitution, insertion and deletion operations needed to transform a hypothesized sentence to the ground truth. \( N \) is the number of words in ground truth.

Implementation Details. In our experiments, videos are divided into 8-frame clips with 50% overlap, with frames cropped and resized to 224 × 224. The output of our 3D-ResNet is a 512-dimensional vector, which represents the clip in sign video. We employ the Adam [30] optimization algorithm for the neural network training. In the Transformer, we apply dropout [31] to the output of each sub-layer, before it is added to the sub-layer input and normalized. Moreover, we apply dropout to the sums of the embeddings and the positional encodings in both the encoder and the decoder stacks. The dropout rate is set to 0.3. In addition, we employ label smoothing [32] with value \( \epsilon_{ls} = 0.2 \). This hurts perplexity, as the model learns to be more unsure, but improves the performance apparently.

4.2. Comparison with the State-of-the-art

In the subsection, we evaluate the performance of our method by comparing it to some existing algorithms on the RWTH-PHOENIX-Weather dataset. The results are summarized in Table 2. In this table, “ins” and “del” mean the average operations of “insertion” and “deletion” that transform the generated sentences into the sentences of ground-truth.

Compared with other methods, our SL-based model achieves a competitive performance with a lower value of “ins” and “del”, which means that the “sub” (“substitution”) is higher. This is due to the fact that the encoder component of the Transformer has the capability to distinguish sign language signal from a sequence of video clip features exactly since the Transformer is based solely on attention mechanisms. However, the translation results do not exhibit a higher level of translation. It means that the decoder component of the Transformer does not complete the translation task perfectly regardless of the limitations of the Transformer or the accuracy of the features of video clips.

As revealed from the results, our RL-based method achieves comparable performance. It’s worth mentioning that our model achieves the best performance on the metrics of “del” and “ins”. The attention mechanism of the Transformer is of great benefit to distinguish effective sign language signal from a sequence of video clip features.

Table 2. Performance of the proposed method and some existing algorithms on RWTH-PHOENIX-Weather. “SL” represents that the model is trained by supervised learning. “RL” represents that we fine-tune the model by reinforcement learning.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Dev(%)</th>
<th>Test(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>del/ins</td>
<td>del/ins</td>
</tr>
<tr>
<td>Deep Hand [33]</td>
<td>16.3/6.4</td>
<td>15.2/4.6</td>
</tr>
<tr>
<td>SubUnet [19]</td>
<td>14.6/4.0</td>
<td>14.3/4.0</td>
</tr>
<tr>
<td>Deep Sign [34]</td>
<td>12.6/5.1</td>
<td>11.1/5.7</td>
</tr>
<tr>
<td>Recurrent CNN [20]</td>
<td>13.7/7.3</td>
<td>12.2/7.5</td>
</tr>
<tr>
<td>CNN-DCN [9]</td>
<td>8.3/4.8</td>
<td>7.6/4.8</td>
</tr>
<tr>
<td>LS-HAN [18]</td>
<td></td>
<td>37.3</td>
</tr>
</tbody>
</table>

Ours (SL)    | 5.7/6.8| 39.7 |
Ours (RL)    | 7.3/5.2| 38.0 |

5. CONCLUSION

In this paper, we propose a deep learning framework composed of 3D-ResNet and the Transformer for continuous sign language recognition (CSLR). Besides, a policy-gradient reinforcement learning (RL) method, which is equipped with a baseline to reduce variance, is utilized to train our end-to-end system directly on the non-differentiable metrics, i.e., word error rate (WER), and leads to performance gain on RWTH-PHOENIX-Weather dataset.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on June 30,2021 at 09:27:08 UTC from IEEE Xplore. Restrictions apply.
6. REFERENCES


