An Improved Deliberation Network with Text Pre-training for Code-Switching Automatic Speech Recognition

Zhijie Shen, Wu Guo

The Department of Electronic Engineering and Information Science (EEIS)
University of Science and Technology of China, Hefei, China

Abstract

This paper proposes an improved deliberation network (DN) for end-to-end code-switching (CS) automatic speech recognition (ASR). In a conventional DN, acoustic encoding and first-pass hypothesis encoding are utilized separately and are simply combined by summation, which cannot take full advantage of their potential complementarity. Hence, the proposed improved DN model exploits the relationship between the two encodings through a two-staged process. First, by integrating the two encodings into a unified semantic space through a shared encoder, and second, by capturing the relevant information from the acoustic encoding through an attention mechanism before the final decoding process. Moreover, the lack of paired training data restricts the generalization ability of the model in CS ASR. To address this problem, the developed DN is pre-trained based on a denoising sequence-to-sequence (seq2seq) objective using unpaired text data. Experiments on a Chinese-English CS dataset demonstrate the effectiveness of the proposed method. Compared with the conventional DN, a 13.5% relative error rate reduction is observed.

Index Terms: automatic speech recognition, code-switching, deliberation network, text pre-training

1. Introduction

Code-switching (CS) is the alternation of languages within a conversation or utterance, and it is a common linguistic phenomenon in multilingual communities across the world [1]. Recently, end-to-end (E2E) automatic speech recognition (ASR) has gained popularity due to its simplicity and effectiveness [2, 3, 4, 5, 6]. Considering CS ASR, the E2E approaches have also attracted great interest [7, 8, 9, 10, 11]. Despite the remarkable success, the current E2E ASR accuracy is still unsatisfactory under noisy or low-resourced situations. A Deliberation Network (DN) is proposed to further improve the E2E ASR performance [12, 13, 14, 15]. The DN model comprises two major parts: the first-pass model and the deliberation re-scorer. The former is an Attention-based Encoder-Decoder (AED) model that produces the first-pass hypothesis for re-scoring. The latter (re-scorer) is also an AED model, which attends to the first-pass hypothesis and the acoustic encoding from the first-pass encoder to perform second-pass decoding. In [13], an additional encoder was introduced to narrow the gap between the re-scorer and the first-pass model. This work also leveraged minimum WER (MWER) training [16] and joint-training that further improved quality, outperforming the text-only re-scoring methods. Moreover, [14] proposed a transformer-based DN by using transformer layers for deliberation re-scoring and a generalized "encoder-decoder" attention mechanism attending to multiple sources.

However, the conventional and the transformer-based DN suffer from several disadvantages. First, the inputs of the re-scorer originate from two different modalities, leading to a considerable discrepancy between their distributions but the conventional DN simply combines them by summation, which is sub-optimal mathematically. Second, the two encodings correspond to a speech-text pair, and therefore an intrinsic relevance between them exists. However, the re-scorer attends to them separately, ignoring the relevance of the two modalities [17]. Hence, this work proposes an improved DN to address the above problems. Specifically, a shared transformer encoder is designed to integrate different modalities into one unified semantic space, and an attention mechanism is applied to these two encodings to exploit the relationship.

One of the significant challenges for CS ASR is data scarcity. In general, an all- neural E2E ASR requires massive amounts of data to train models [6], with data collection being time-consuming and expensive. Several attempts have been made to alleviate this problem. [18] and [19] leveraged the Text-To-Speech (TTS) system to generate new speech corresponding to CS texts. Moreover, [18] used audio splicing to obtain the individual language-dependent segments, which were then randomly spliced to generate new CS utterances. In [20] and [21], the models were pre-trained using monolingual speech-transcript pairs for initialization, and then the knowledge was transferred to CS. Significant improvements were observed with the help of paired synthesized CS data or monolingual data. However, these approaches require speech-transcript pairs as training data and cannot utilize text-only data. Therefore, this work explores utilizing unpaired text data to improve performance, which is easier to obtain than speech. Both monolingual and CS texts are used. The denoising method of BART [22] is employed to pre-train the re-scorer, where the model tries to reconstruct the corrupted inputs based on a sequence-to-sequence (seq2seq) framework. We carried out our experiments using IFLYTEK datasets of Chinese-English CS and the results demonstrate that the proposed methods are highly effective, yielding up to 13.5% relative error rate reduction.

The remainder of this paper is organized as follows. Section 2 describes the methods in detail, including the baseline DN model, the improved DN model, and the text pre-training method. Section 3 introduces our experimental setups, including the dataset analysis, the model architecture, the training configurations, and the evaluation metric. Section 4 presents the experimental results and analyzes them in detail while Section 5 concludes this work.

2. Methods

This section introduces the baseline DN model, the improved DN model, and the text pre-training approach.
2.1. Baseline DN for ASR

The baseline DN is a typical two-pass ASR model [13] comprising a first-pass model and a deliberation re-scorer (Figure 1(a)). The first-pass model is an AED model, where the input log-mel filterbank features \( x = (x_1, x_2, \ldots, x_T) \) are first down-sampled and then input into the encoder to obtain the acoustic encoding \( h^a = (h^a_1, h^a_2, \ldots, h^a_T) \), where \( T \) and \( T' \) are the number of frames before and after down-sampling respectively. The acoustic encoding \( h^a \) is the input of the attention-based decoder. Finally, the decoder outputs the first-pass hypothesis \( y_1 = (y_1, y_2, \ldots, y_L) \), where \( L \) is the length. The deliberation re-scorer is also an AED model but contains two attention inputs. This model first encodes the first-pass hypothesis by a multi-layer transformer [23], denoted as \( h^b = (h^b_1, h^b_2, \ldots, h^b_L) \). Then, the attention mechanism is applied to both the acoustic encoding \( h^a \) and the first-pass hypothesis encoding \( h^b \) to obtain the context vectors \( c^a \) and \( c^b \) respectively. Note that an optional additional encoder can be used to pre-process \( h^a \). Finally, the two context vectors are summed as inputs to the LSTM decoder to produce the second-pass hypothesis \( y_2 \). In this work, we employ summation instead of concatenation [13] because we found summation leads to higher recognition accuracy.

2.2. Improved DN

In this sub-section, we introduce two approaches designed for improving the baseline DN model, with the improved DN architecture illustrated in Figure 1(b).

2.2.1. Unified mapping

In the baseline DN, the deliberation re-scorer attends to acoustic encoding \( h^a \) and first-pass hypothesis encoding \( h^b \) separately. Nevertheless, in the improved DN, \( h^a \) and \( h^b \) are integrated into a unified semantic space by a shared transformer encoder, making it more effective to merge the two context vectors \( c^a \) and \( c^b \). This approach brings another advantage that the shared encoder plays the role of an additional encoder for acoustic encoding, which has been proven effective in [13]. We will conduct a comparative study between the two methods in sub-section 4.1.

2.2.2. Encoding refinement

Since the two encodings \( h^a \) and \( h^b \) mentioned above correspond to a speech-text pair, intrinsic relevance can be exploited. Consequently, we propose an encoding refinement approach to effectively fuse the two encodings. Specifically, we first let \( h^h \) perform a cross-attention query operation on \( h^a \), to generate the context vector \( h^r \), which is a content-based attention [24]:

\[ \omega_{i,k} = \text{Attention}(s_{i-1}, h^a_k) \]

\[ h^r_i = \sum_{k=1}^{T'} \omega_{i,k} h^a_k \]

\[ s_i = \text{LSTM}(h^r_i, h^a_{i-1}) \]

where \( \omega_{i,k} \) is the attention weight corresponding to \( h^a_k \) and \( h^r_i \). \( s_i \) is the hidden state produced by an LSTM layer and \( h^r = (h^r_1, \ldots, h^r_L) \) is the context vector, which is a weighted average of the original acoustic encoding \( h^a \). Therefore, \( h^r \) can be regarded as a refined version of \( h^a \). Moreover, the first-pass hypothesis encoding \( h^b \) is concatenated with \( h^r \) in time. Finally, using a self-attention mechanism, the duplicated or redundant time spans of the two encodings are summarized, emphasizing the important ones. The skip connection is applied to the self-attention mechanism:

\[ p = (h^1, \ldots, h^L, h^r_1, \ldots, h^r_L) \]

\[ \omega_{i,k} = \text{Attention}(s_{i-1}, p_k) \]

\[ h^r_i = \sum_{k=1}^{2L} \omega_{i,k} p_k + p_i \]

\[ s_i = \text{LSTM}(h^r_i, p_{i-1}) \]

where \( L \) is the length of \( h^h \) and \( h^r \), and \( h^* = (h^1, \ldots, h_{2L}) \) is the final encoding refinement output.

2.3. Text Pre-training

For both the baseline and the improved DNs, the deliberation re-scorer takes not only acoustics but also first-pass hypotheses as inputs for second-pass decoding. Therefore, the re-scorer can be regarded as an implicit language model. We propose to pre-train the re-scorer using the denoising seq2seq method, which...
is closely related to BART [22]. This method allows us to use external text data, which is helpful since CS speech-transcript data is scarce. As illustrated in Figure 2, only the re-scorer’s top part (text part) parameters of the re-scorer are updated in the pre-training process.

Next, we introduce the pre-training procedure of our method. In short, a certain proportion of the input tokens are randomly replaced with mask symbols, and the model tries to reconstruct the original texts based on a sequence-to-sequence framework. Specifically, the transformer encoder is first used to obtain the corrupted input encoding, which is then input into the attention model. Finally, the LSTM decoder outputs the likelihood of the original sequence. The cross-entropy loss between the decoder output and the original sequence is calculated to optimize the model.

2.3.1. Fine-tuning

In the fine-tuning procedure, we replace the pre-trained embedding layer with an embedding layer followed by two encoder layers, the parameters of which are randomly initialized. The new encoder maps the first-pass hypothesis into the corrupted text embedding used in pre-training [22]. In this work, the pre-trained encoder has two layers, and the baseline encoder has four layers so the number of the parameters is equal.

By incorporating the re-scorer pre-training step into the entire training process, we expand the three steps of [25] with the detailed ‘pre-training + fine-tuning’ procedure being as follows:

1. Pre-train the re-scorer with external texts.
2. Train the first-pass AED model for ASR.
3. Utilize the two models in steps (1) and (2), freeze the first-pass model, and update only the re-scorer for ASR.
4. Jointly update the two models with the combined loss:

\[
L_{\text{joint}} = L_{CE}^1(\theta_e, \theta_r) + \alpha L_{CE}^d(\theta_e, \theta_r) \tag{8}
\]

where \(L_{CE}^1(\cdot)\) and \(L_{CE}^d(\cdot)\) are the cross-entropy losses of the first-pass model and the deliberation re-scorer, respectively. \(\theta_e\) denote the parameters of the first-pass encoder, \(\theta_r\) refer to the parameters of the attention model and LSTM decoder in the first-pass model, and \(\theta_r\) are the parameters of the encoder, attention model, and the LSTM decoder in the re-scorer.

3. Experimental setup

3.1. Datasets

Our work focuses on Chinese-English CS ASR, and evaluates the proposed method on the IFLYTEK CS dataset, comprising approximately 101.5-hour speech data. For the ASR training, the dataset consists of approximately 23.6h monolingual Chinese data, 12.6h monolingual English data, and 65.3h CS data. Our work aims to improve the recognition accuracy for CS speech, and therefore, the test set contains only 6.4h CS data, and no monolingual data is used. All speech data is collected with a 16kHz sampling rate. Moreover, approximately 3.0M utterances are used during pre-training. These utterances are obtained by pooling monolingual texts and CS texts from external IFLYTEK datasets.

The statistics of the pre-training, ASR training, and test sets are reported in Table 1 and Table 2, employing four metrics to measure the CS data complexity: percentage of English tokens per utterance, average switches per utterance, Switching-Point Fraction (SPF) [26], and Code-Mixing Index (CMI) [27]. We calculate the metrics by averaging the values for all CS utterances. As reported in Table 3, the ratio of Chinese to English is about 5:1 in the CS data.

### Table 1: Statistics of the pre-training set. ‘ZH’, ‘EN’, and ‘CS’ refer to Chinese, English, and code-switching, respectively.

<table>
<thead>
<tr>
<th></th>
<th>ALL</th>
<th>ZH</th>
<th>EN</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Utterances</td>
<td>3.0M</td>
<td>41.6%</td>
<td>39.4%</td>
<td>19.0%</td>
</tr>
<tr>
<td>#Tokens</td>
<td>66.1M</td>
<td>56.4%</td>
<td>43.6%</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 2: Statistics of the ASR training and test set.

<table>
<thead>
<tr>
<th></th>
<th>ALL</th>
<th>ZH</th>
<th>EN</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train Duration</td>
<td>101.5Hr</td>
<td>23.2%</td>
<td>12.4%</td>
<td>64.4%</td>
</tr>
<tr>
<td>#Utterances</td>
<td>120.1K</td>
<td>26.6%</td>
<td>16.2%</td>
<td>57.2%</td>
</tr>
<tr>
<td>#Tokens</td>
<td>1.5M</td>
<td>78.2%</td>
<td>21.8%</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table 3: CS analysis for the pre-training set, the ASR training set, and the ASR test set.

<table>
<thead>
<tr>
<th></th>
<th>Pre-train</th>
<th>ASR Train</th>
<th>ASR Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage of EN</td>
<td>16.56%</td>
<td>17.21%</td>
<td>17.26%</td>
</tr>
<tr>
<td>Average switches</td>
<td>1.84</td>
<td>1.81</td>
<td>1.81</td>
</tr>
<tr>
<td>SPF</td>
<td>0.13</td>
<td>0.14</td>
<td>0.15</td>
</tr>
<tr>
<td>CMI</td>
<td>0.28</td>
<td>0.31</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Table 4: TERs (%) of different ASR systems. ‘PT’ denotes whether the re-scorer is pre-trained or not. TER is calculated on the entire sequence (ALL), on the Chinese parts (ZH), and on the English parts (EN), respectively. The additional encoder is not applied.

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>PT</th>
<th>ALL</th>
<th>ZH</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baseline DN</td>
<td>no</td>
<td>32.6</td>
<td>31.2</td>
<td>49.6</td>
</tr>
<tr>
<td>2</td>
<td>+ Unified mapping</td>
<td>no</td>
<td>31.7</td>
<td>30.4</td>
<td>48.0</td>
</tr>
<tr>
<td>3</td>
<td>+ Encoding refinement</td>
<td>no</td>
<td>30.4</td>
<td>29.1</td>
<td>45.8</td>
</tr>
<tr>
<td>4</td>
<td>+ Text pre-training</td>
<td>yes</td>
<td>31.1</td>
<td>29.9</td>
<td>46.6</td>
</tr>
<tr>
<td>5</td>
<td>+ All proposed methods</td>
<td>yes</td>
<td>28.2</td>
<td>27.1</td>
<td>42.6</td>
</tr>
</tbody>
</table>

Table 5: TERs (%) with the unified mapping or the additional encoder.

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>ALL</th>
<th>ZH</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Unified mapping</td>
<td>31.7</td>
<td>30.4</td>
<td>48.0</td>
</tr>
<tr>
<td>2</td>
<td>Additional encoder</td>
<td>31.9</td>
<td>30.6</td>
<td>48.1</td>
</tr>
</tbody>
</table>

4. Results

Table 4 presents the TER performance of the baseline DN, the improved DN with or without pre-training.

**Improved DN:** The overall TER of the model employing unified mapping (row 2) is 31.7%, which is 0.9% (absolute) lower than the baseline DN (row 1). Moreover, this method can be effectively combined with the encoding refinement approach (row 3). Compared with the baseline DN, the overall TER is reduced by 6.7% (relative).

**Unpaired text pre-training:** The pre-training method is applied over the baseline and the improved DN models, leading to consistent improvements for both of them. For example, considering the baseline DN, the overall TER is reduced by 1.5% (absolute). Additionally, the results reveal that the pre-training method improves English recognition more than Chinese, with the relative TER reduction being 6.0% and 4.2%, respectively.

Table 6: TERs (%) with different methods to merge $h'$ and $h^b$.

<table>
<thead>
<tr>
<th>No.</th>
<th>Method</th>
<th>ALL</th>
<th>ZH</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Concatenation in time</td>
<td>28.2</td>
<td>27.1</td>
<td>42.6</td>
</tr>
<tr>
<td>2</td>
<td>Sum</td>
<td>30.2</td>
<td>29.0</td>
<td>45.1</td>
</tr>
<tr>
<td>3</td>
<td>Concatenation in feature</td>
<td>32.7</td>
<td>31.5</td>
<td>49.1</td>
</tr>
<tr>
<td>4</td>
<td>Gated averaging</td>
<td>32.2</td>
<td>30.1</td>
<td>51.0</td>
</tr>
</tbody>
</table>

The reason is that the amount of English data (12.6h) is far less than Chinese data (23.6h), and thus external text data are more significant in English recognition. Furthermore, by merging all the proposed methods (unified mapping + encoding refinement + pre-training), an overall TER reduction of 13.5% (relative) is obtained, in contrast with the baseline DN.

4.1. Unified mapping vs. Additional encoder

As described in 2.2.2, the unified mapping method introduces a shared transformer encoder, which plays the role of the additional encoder proposed in [13]. We now compare the performance of the two methods. The corresponding results are reported in Table 5, revealing that our method (row 1) achieves an overall TER reduction of 0.2% (absolute) compared with the additional encoder method (row 2).

4.2. Comparison with ‘Concatenation in time’

In the encoding refinement method, $h'$ is concatenated with $h^b$ in time. We then conduct experiments that employ different methods to merge $h'$ and $h^b$, namely, sum, concatenation followed by projection, and gated averaging. In the concatenation case, the vectors are concatenated in the feature dimension (results in 512 dimensions) and projected back to the original dimension (256). The gated averaging is similar to that in [32]. The corresponding results are listed in Table 6, revealing that substituting the ‘Concatenation in time’ module with any other module leads to significant performance degradation. This is because the ‘Concatenation in time’ method collaborates well with the self-attention mechanism, allowing the self-attention mechanism to generate context by learning long-range dependencies between $h'$ and $h^b$.

5. Conclusions

This work proposes an improved DN for CS ASR by developing a unified mapping and an encoding refinement scheme. We also exploit the denoising seq2seq objective to pre-train the deliberation re-scorer with external text data. Our best model achieves a TER of 28.2% on a Chinese-English CS test set, which outperforms the baseline model by 13.5%. We also observe that English recognition benefits more from pre-training than Chinese because English data is scarcer in our work. Moreover, we conduct several comparative studies which prove the effectiveness of our method.

Future work will consider different text pre-training strategies to initialize the deliberation re-scorer. We will also investigate the ensemble methods and the joint decoding algorithms to combine both the first-pass and second-pass hypotheses.

6. Acknowledgements

This work was partially funded by the National Natural Science Foundation of China (Grant No. U1836219).
7. References


