Abstract

Attention-based sequence-to-sequence (seq2seq) models have achieved promising results in automatic speech recognition (ASR). However, as these models decode in a left-to-right way, they do not have access to context on the right. We leverage both left and right context by applying BERT as an external language model to seq2seq ASR through knowledge distillation. In our proposed method, BERT generates soft labels to guide the training of seq2seq ASR. Furthermore, we leverage context beyond the current utterance as input to BERT. Experimental evaluations show that our method significantly improves the ASR performance from the seq2seq baseline on the Corpus of Spontaneous Japanese (CSJ). Knowledge distillation from BERT outperforms that from a transformer LM that only looks at left context. We also show the effectiveness of leveraging context beyond the current utterance. Our method outperforms other LM application approaches such as n-best rescoring and shallow fusion, while it does not require extra inference cost. 

Index Terms: speech recognition, sequence-to-sequence models, language model, BERT, knowledge distillation

1. Introduction

End-to-end models that directly map acoustic features into symbol sequences have shown promising results in automatic speech recognition (ASR). Compared to conventional DNN-HMM hybrid systems, end-to-end models have the advantages of a simplified architecture and fast decoding. There are various choices when it comes to end-to-end models: connectionist temporal classification (CTC) [1], attention-based sequence-to-sequence (seq2seq) models [2,3], and RNN-transducer models [4,5]. In this study, we adopt attention-based seq2seq models.

Seq2seq ASR models use paired speech and text for training. In addition, unpaired text that is more readily available can be used to improve them. An external language model (LM) is trained separately on unpaired text, and various approaches for applying the LM to ASR have been proposed. In n-best rescoring, n-best hypotheses are obtained from ASR, followed by the addition of their LM scores, and then the best-scored hypothesi among them is selected. Language model fusion approaches such as shallow fusion [6], deep fusion [7], and cold fusion [8,9] utilize an external LM during beam-search decoding. In shallow fusion, the linearly interpolated score from both the LM and the ASR model is used in beam search during the inference stage. More recently, knowledge distillation [10] -based LM integration has been proposed [11]. In this approach, the LM (teacher model) provides soft labels to guide the seq2seq model (student model) training. The LM is used during the training stage but is not required during the inference stage.

In the above-mentioned approaches that apply LM to seq2seq ASR, n-gram, RNNLM, or transformer [12] LM is conventionally used. We call them “unidirectional” LMs, which predict each word on the basis of its left context. In this study, we propose to apply BERT [13] as an external LM. BERT features Masked Language Modeling (MLM) in the pre-training objective, where MLM masks a word from the input and then predicts the original word. BERT can be called a “bidirectional” LM that predicts each word on the basis of both its left and right context.

Seq2seq models decode in a left-to-right way, and therefore they do not have access to the right context during training or inference. We aim to alleviate this seq2seq’s left-to-right bias, by taking advantage of BERT’s bidirectional nature. N-best rescoring with BERT was proposed in [14,15], but the recognition result was restricted to hypotheses from left-to-right decoding. On the other hand, BERT is difficult to use in LM fusion approaches because right (future) context that has not yet been decoded cannot be accessed during inference. To solve these issues, we propose to apply BERT to ASR through knowledge distillation. BERT (teacher model) provides soft labels using both left and right contexts of a current utterance for the seq2seq model (student model) training. Furthermore, we propose to use not only right context but also context beyond utterance boundaries during the training stage. In spontaneous ASR tasks such as presentation and conversation, the speech comprises a series of utterances. In our proposed method, previous utterances, the current utterance, and future utterances are concatenated up to the fixed length of tokens and then fed into BERT. BERT provides soft labels based on context that spans across utterances, which helps achieve better seq2seq ASR training.

2. Preliminaries and related work

2.1. Sequence-to-sequence ASR

In attention-based seq2seq ASR, we model the mapping between acoustic features and symbol sequences using two distinct networks. One is an encoder network that transforms acoustic features into a high-level representation. The other is a decoder network that predicts a sequence of symbols using the encoded representation. At each decoding step, the decoder predicts a symbol using a relevant portion of the encoded representation and previously decoded symbols. In this study, we implemented the encoder with a multi-layer bidirectional LSTM and the decoder with a unidirectional LSTM.

Let \( X = (x_1, \ldots, x_T) \) denote a sequence of input acoustic features. Let \( y = (y_1, \ldots, y_N) \) denote a sequence of target symbols. The target symbols are subwords in this study, and \( y_i \in \{1, \ldots, V\} \), where \( V \) denotes the vocabulary size. We define the seq2seq model’s output probability of subword \( v \) for the \( i \)-th target as

\[
P_{ASR}^{(i)}(v) = p(v | X, y_{<i})
\]

\( y_{<i} \) denotes the left context of \( y_i \), that is \( y_{<i} = (y_1, \ldots, y_{i-1}) \). During the training of a seq2seq model, we minimize the fol-
lowing cross-entropy objective:

\[
\mathcal{L}_{ASR} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{v=1}^{V} \delta(v, y_i) \log P^{(v)}_{ASR}(y_i)
\]  

(2)

where \(\delta(v, y_i)\) becomes 1 when \(v = y_i\), and 0 otherwise.

2.2. BERT

BERT [13] is a mechanism for LM pre-training that consists of a
multi-layer bidirectional transformer encoder [12]. BERT can
be pre-trained on a large unlabeled text and then be fine-tuned
on a limited labeled text. It has shown excellent results in many
downstream natural language processing tasks. BERT’s success
comes from learning “deep bidirectional” representations. Pre-
vio\(\)us approaches to LM pre-training such as OpenAI GPT [16]
(unidirectional) and ELMo [17] (shallow concatenation of
left-to-right and right-to-left RNNLMs) do not perform as well as
BERT because they are not “deeply bidirectional”.

BERT originally has two pre-training objectives: Masked
Language Modeling (MLM) and Next Sentence Prediction
(NSP). MLM randomly replaces some of the input tokens with
\([\text{MASK}]\) tokens and then predicts the original word on the
basis of its both left and right context. NSP predicts whether two
input sentences appear consecutively in a corpus to model sen-
tence relationships.

2.3. Bidirectional context in seq2seq models

Seq2seq models predict each word using its left context. Due to
this autoregressive property, it is difficult for seq2seq models to
leverage right context during the training and inference stages.
In seq2seq decoding, later predictions depend on the accuracy
of previous predictions, and therefore the issue of error accumu-
lation arises [18]. Previous studies have addressed this issue by
using right context in seq2seq ASR [19] and neural machine
translation (NMT) [20] [21]. In [19], a left-to-right and a right-
to-left decoder generate \(n\)-best hypotheses respectively, and the
two \(n\)-best hypotheses are then concatenated to make new hy-
potheses. In [20], a second-pass deliberation decoder that can
leverage right context was proposed. Synchronous bidirectional
decoding in a single model was proposed in [21].

Meanwhile, some studies have leveraged right context dur-
ding the seq2seq model training by distilling the knowledge of a
“bidirectional” teacher model [22] [23]. In [22], which is a suc-
cceeding work of [11], Causal cloZe completeR (COR) was
proposed to model both left and right context within an utter-
ance. In COR, the output of a stack of left-to-right transformer
blocks and a stack of right-to-left ones are concatenated and
fed into a subsequent fusion transformer block. Compared to
BERT, it only performs shallow concatenation of two directions
of transformer blocks, and as such is not “deeply bidirectional”.
On the other hand, we adopt BERT, which has a simpler and
more general architecture. Furthermore, we use context that
spans across utterances as input to BERT for better distillation,
whereas the context is limited to the current utterance in [23].
In [23], a source sequence of tokens and a target sequence of
tokens are fed into BERT to generate soft labels for text-to-text
transduction tasks such as NMT.

2.4. Context beyond utterance boundaries in ASR

ASR is typically done at the utterance level, but context infor-
mation beyond the utterance level can help improve seq2seg
ASR [24] [25] [26]. A context vector generated from the previous
utterance is incorporated into the decoder state in the current
utterance in these studies.

In our method, context information beyond the utterance
level is not incorporated into the ASR decoder but fed into
BERT to predict better soft labels for ASR. During inference,
BERT is not used, and therefore our method does not add any
extra procedure or component to utterance-level seq2seq ASR.

3. Proposed method

3.1. Pre-training BERT

In our proposed method, BERT is used as an external LM that
predicts a masked word based on its context. We need the MLM
pre-training objective itself, and therefore fine-tuning for down-
stream tasks is not conducted. NSP is also removed from the
pre-training objective. Following RoBERTa [27], BERT’s in-
put is packed with full-length sequences sampled contiguously
from the corpus.

3.2. Distilling the knowledge of BERT

In our knowledge distillation, BERT serves as a teacher model
and a seq2seg ASR model serves as a student model. Pre-
trained BERT provides soft labels to guide seq2seq ASR training.
These soft labels encourage the seq2seg ASR model to generate
more syntactically or semantically likely results.

Seq2seq ASR training with the knowledge of BERT is for-
mulated as follows. The speech in the corpus is split into a
series of utterances, and the ASR model is trained on utterance-
level data. As in Section 2.1, \(X\) denotes acoustic features
in an utterance, and \(y\) denotes a label sequence correspond-
ing to \(X\). We utilize context beyond the current utterance as
input to BERT in our method. Let \(y^{(L)} = (y_1^{(L)},...,y_N^{(L)})\)
be a subword sequence for previous (left) utterances and
\(y^{(R)} = (y_1^{(R)},...,y_N^{(R)})\) one for future (right) utterances.
The length of \(y^{(L)} = L\) and that of \(y^{(R)} = R\) are decided such
that the sum of \(L, R,\) and \(N\) (the label length of the current
utterance) is constant (e.g. \(L + R + N = 256\)) and that \(L \neq R\) and
\(R\) are the same (i.e. \(L = R\)).

We define BERT’s output probability of subword \(v\) for the
\(i\)-th target label as

\[
P^{(i)}_{BERT}(v) = p(v | [y^{(L)}; y_1; y^{(R)}]) = \frac{\exp(z_i/T)}{\sum_{j=1}^{V} \exp(z_j/T)}
\]

(3)

where \(z_i\) is a logit before the softmax layer and \(T\) is a tem-
oxperature parameter. We obtain \(y_j\) by converting the \(i\)-th token to
\([\text{MASK}]\) that is, \(y_j = (y_1,\ldots,y_i,[\text{MASK}],y_{i+1},\ldots,y_N)\),
y_1 is concatenated with y^{(L)} and y^{(R)}, then fed into BERT as
\([y^{(L)}; y_1; y^{(R)}]\).

Let \(P_{ASR}^{(i)}\) and \(P_{BERT}^{(i)}\) denote the probability distribu-
tion for the \(i\)-th target predicted by a seq2seg ASR model and by
BERT, respectively. Our goal here is to distill the knowledge
of BERT and transfer it to the seq2seg ASR model by making
\(P_{ASR}^{(i)}\) close to \(P_{BERT}^{(i)}\), as illustrated in Figure 1. Thus,
we minimize the Kullback-Leibler (KL) divergence between
\(P_{ASR}^{(i)}\) and \(P_{BERT}^{(i)}\) for each \(i\).

\[
KL(P_{BERT}^{(i)} || P_{ASR}^{(i)}) = -\sum_{v=1}^{V} P_{BERT}^{(i,v)} \log \frac{P_{ASR}^{(i,v)}}{P_{BERT}^{(i)}}
\]

(5)

\(P_{BERT}^{(i)}\) is fixed during distillation, and therefore minimizing
the KL divergence over the sequence is equivalent to mini-
Figure 1: Illustration of our proposed method. BERT generates the soft label (= $P_{BERT}^{(3)}$) using context in which $y_3$ is masked and the current utterance, previous and future utterances are concatenated. The target label for $P_{ASR}^{(3)}$ is given by not only the hard label (= $y_3$) but also the soft label (= $P_{BERT}^{(3)}$).

4. Experimental evaluations

4.1. Experimental conditions

We evaluated our method using the Corpus of Spontaneous Japanese (CSJ) \[29\] and the Balanced Corpus of Contemporary Written Japanese (BCCWJ) \[30\]. CSJ includes two subcorpora, CSJ-APS and CSJ-SPS. CSJ-APS consists of about 240 hours of oral presentation speeches from academic meetings, and CSJ-SPS consists of about 280 hours of simulated presentation speeches on general topics. CSJ-eva1, which is an official test set of CSJ-APS, was used for evaluation. We also used BCCWJ-PB and BCCWJ-LB in BCCWJ as additional text for training LMs. BCCWJ-PB consists of samples extracted from published books, and BCCWJ-LB consists of samples from books registered in libraries. The text is tokenized using Byte Pair Encoding \[31\] of vocabulary size 7520. BCCWJ-PB and BCCWJ-LB have about 37M and 40M subword tokens, respectively. The transcriptions of CSJ-APS and CSJ-SPS have about 3.9M and 4.1M subword tokens, respectively.

In our seq2seq ASR, the encoder consists of 5 layers of bidirectional LSTMs with 320 hidden states, and the decoder consists of a single LSTM layer with 320 hidden states. We trained the seq2seq model on CSJ-APS with a batch size of 25 utterances. The average token length of utterances was about 24 (maximum: 118, minimum: 1). We used Adam \[32\] with the learning rate of 1e-4 for optimizing the ASR model. SpecAugment \[33\] was applied to the acoustic features. We also applied label smoothing \[34\]. In target labels, the probability of 0.1 was distributed uniformly over all classes. In decoding, we used beam search with a beam width of 5.

We trained BERT and a unidirectional transformer LM for comparison. BERT and the transformer LM have 6 layers of transformer blocks with 512 hidden states and 8 attention heads. We trained them on BCCWJ-PB and BCCWJ-LB first, then on the transcriptions of CSJ-APS and CSJ-SPS. We sampled 150 sequences of length 256 for each pre-training step. In BERT, we randomly selected 8% of the tokens in each sequence and replaced them with [MASK] tokens. In CSJ, we used Adam with the learning rate of 1e-4 and linear decay for the first 10% of total steps and linear decay. K for top-K distillation was set to 8 in all our experiments. The temperature parameter T in Eq. (4) and the distillation weight $\alpha$ in Eq. (7) were adjusted using the development set. Our code for the proposed method is available\[1\].

\[1\]https://github.com/hfutami/distill-bert-for-seq2seq-asr
Table 1: The performance for ASR trained on CSJ-APS (240h) with knowledge distillation-based LM integration. “TrfLM(uni)” in the “LM” column denotes the transformer LM.

| LM          | Context size | WER(%) |
|-------------|--------------|--------|
| —           | —            | 10.31  |
| TrfLM(uni)  | utterance    | 9.89   |
| TrfLM(uni)  | 256          | 10.01  |
| BERT        | utterance    | 9.53   |
| BERT        | 256          | 9.19   |

Table 2: Ablation studies on the length of BERT’s input during pre-training and distillation.

| Context size | Pre-training | Distillation | WER(%) |
|--------------|--------------|--------------|--------|
| 64           | utterance    | 9.91         |
| 64           | 64           | 9.60         |
| 128          | utterance    | 9.62         |
| 128          | 128          | 9.40         |
| 256          | utterance    | 9.53         |
| 256          | 64           | 9.28         |
| 256          | 128          | 9.28         |
| 256          | 256          | 9.19         |

4.2. Experimental results

We evaluated our method through ASR experiments. First, we compared the performances of the ASR models trained using BERT and the unidirectional transformer LM (TrfLM(uni)) as a teacher model. We also evaluated the effectiveness of using context beyond the current utterance. The ASR results are shown in Table 1. The result denoted as “utterance” in the “Context size” column corresponds to the ASR model guided by soft labels based on context within the current utterance. The result denoted as “256” in the “Context size” column corresponds to that guided by soft labels based on context of length 256 that spans across utterances. In TrfLM(uni), we added only previous utterances to the current utterance as context. The first line in the table denotes the baseline ASR without distillation. As shown in Table 1 knowledge distillation-based LM integration consistently improved the performance of the ASR model. We found that distillation from BERT outperformed that from the TrfLM(uni), which indicates the effectiveness of leveraging not only left context but also right context that seq2seq ASR cannot utilize. We also found that incorporating context beyond the current utterance was important for distillation from BERT by comparing line 4 and 5 in the table. This result improved the WER by 10.80% relatively over the baseline.

Next, we compared our method with two other LM application approaches. Shallow fusion (SF) and n-best rescoring were applied to the baseline and were compared to the ASR model trained with our method (the last line in Table 1). As shown in Figure 1, our method outperformed both shallow fusion and n-best rescoring regardless of the beam width. We also applied shallow fusion and n-best rescoring to the ASR model trained through our method and obtained some improvements, which were not as large as those applied to the baseline. This can be interpreted as the ASR model with our method had already learned the effect of applying an external LM through distillation.

Figure 2: Comparisons and combinations with other LM application approaches. “SF” denotes shallow fusion.

Table 3: The performance for ASR trained on an increased amount of data (520h, both CSJ-APS and CSJ-SPS).

| LM          | Context size | WER(%) |
|-------------|--------------|--------|
| —           | —            | 8.43   |
| BERT        | 256          | 7.85   |

Next, we conducted ablation studies on context size during pre-training and distillation. The results are shown in Table 2. We found that the use of longer context in the pre-training led to better ASR performance. We also found that distillation from BERT using the same context size as pre-training performed best.

Finally, to see the effect of an increased amount of training data for ASR in our method, we trained another ASR model on both CSJ-APS and CSJ-SPS (total 520h) and evaluated the performance. As shown in Table 3, our method was still effective for this better baseline ASR model trained on an increased amount of paired data.

5. Conclusions

BERT can be pre-trained on a large unpaired text, and can also leverage not only left context but also right context that seq2seq ASR models do not have access to. In this study, we have proposed a method in which the knowledge of BERT is transferred to seq2seq ASR through a knowledge distillation framework and demonstrated its effectiveness through experiments. We found that distillation from BERT yields better ASR performance than that from the transformer LM. We also found that the knowledge of BERT based on context that spans across utterances further improved the performance of seq2seq ASR. Our proposed method outperformed other LM application approaches such as n-best rescoring and shallow fusion, including rescoring with BERT, even though our method does not require extra inference cost. As a future work, we will investigate applying other LM pre-training mechanisms such as XLNet and ELECTRA to ASR.
