MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Joint CTC/attention decoding for end-to-end speech recognition

Hori, T.; Watanabe, S.; Hershey, J.R.

TR2017-103 July 2017

Abstract

End-to-end automatic speech recognition (ASR) has become a popular alternative to conventional DNN/HMM systems because it avoids the need for linguistic resources such as pronunciation dictionary, tokenization, and contextdependency trees, leading to a greatly simplified model-building process. There are two major types of end-to-end architectures for ASR: attention-based methods use an attention mechanism to perform alignment between acoustic frames and recognized symbols, and connectionist temporal classification (CTC), uses Markov assumptions to efficiently solve sequential problems by dynamic programming. This paper proposes a joint decoding algorithm for end-to-end ASR with a hybrid CTC/attention architecture, which effectively utilizes both advantages in decoding. We have applied the proposed method to two ASR benchmarks (spontaneous Japanese and Mandarin Chinese), and showing the comparable performance to conventional state-of-the-art DNN/HMM ASR systems without linguistic resources.

Association for Computational Linguistics (ACL)

Joint CTC/attention decoding for end-to-end speech recognition

Takaaki Hori, Shinji Watanabe, John R. Hershey
Mitsubishi Electric Research Laboratories (MERL)
{thori,watanabe,hershey}@merl.com

Abstract

End-to-end automatic speech recognition (ASR) has become a popular alternative to conventional DNN/HMM systems because it avoids the need for linguistic resources such as pronunciation dictionary, tokenization, and context-dependency trees, leading to a greatly simplified model-building process. There are two major types of end-to-end architectures for ASR: attention-based methods use an attention mechanism to perform alignment between acoustic frames and recognized symbols, and connectionist temporal classification (CTC), uses Markov assumptions to efficiently solve sequential problems by dynamic programming. This paper proposes a joint decoding algorithm for end-to-end ASR with a hybrid CTC/attention architecture, which effectively utilizes both advantages in decoding. We have applied the proposed method to two ASR benchmarks (spontaneous Japanese and Mandarin Chinese), and showing the comparable performance to conventional state-of-the-art DNN/HMM ASR systems without linguistic resources.

1 Introduction

Automatic speech recognition (ASR) is currently a mature set of technologies that have been widely deployed, resulting in great success in interface applications such as voice search. A typical ASR system is factorized into several modules including acoustic, lexicon, and language models based on a probabilistic noisy channel model (Jelinek, 1976). Over the last decade, dramatic improvements in acoustic and language models have been driven by machine learning techniques known as deep learning (Hinton et al., 2012).

However, current systems lean heavily on the scaffolding of complicated legacy architectures that grew up around traditional techniques. For example, when we build an acoustic model from scratch, we have to first build hidden Markov model (HMM) and Gaussian mixture model (GMM) followed by deep neural networks (DNN). In addition, the factorization of acoustic, lexicon, and language models is derived by conditional independence assumptions (especially Markov assumptions), although the data do not necessarily follow such assumptions leading to model misspecification. This factorization form also yields a local optimum since the above modules are optimized separately. Further, to well factorize acoustic and language models, the system requires linguistic knowledge based on a lexicon model, which is usually based on a hand-crafted pronunciation dictionary to map word to phoneme sequence. In addition to the pronunciation dictionary issue, some languages, which do not explicitly have a word boundary, need language-specific tokenization modules (Kudo et al., 2004; Bird, 2006) for language modeling. Finally, inference/decoding has to be performed by integrating all modules resulting in complex decoding. Consequently, it is quite difficult for non-experts to use/develop ASR systems for new applications, especially for new languages.

End-to-end ASR has the goal of simplifying the above module-based architecture into a single-network architecture within a deep learning framework, in order to address the above issues. There are two major types of end-to-end architectures for ASR: attention-based methods use an attention mechanism to perform alignment between acoustic frames and recognized symbols, and connectionist temporal classification (CTC), uses Markov

assumptions to efficiently solve sequential problems by dynamic programming (Chorowski et al., 2014; Graves and Jaitly, 2014).

The attention-based end-to-end method solves the ASR problem as a sequence mapping from speech feature sequences to text by using encoder-decoder architecture. The decoder network uses an attention mechanism to find an alignment between each element of the output sequence and the hidden states generated by the acoustic encoder network for each frame of acoustic input (Chorowski et al., 2014, 2015; Chan et al., 2015; Lu et al., 2016). This basic temporal attention mechanism is too flexible in the sense that it allows extremely non-sequential alignments. This may be fine for applications such as machine translation where input and output word order are different (Bahdanau et al., 2014; Wu et al., 2016). However, in speech recognition, the feature inputs and corresponding letter outputs generally proceed in the same order. Another problem is that the input and output sequences in ASR can have very different lengths, and these vary greatly from case to case, depending on the speaking rate and writing system, making it more difficult to track the alignment.

However, an advantage is that the attention mechanism does not require any conditional independence assumptions, and could address all the problems cited above. Although the alignment problems of attention-based mechanisms have been partially addressed in (Chorowski et al., 2014; Chorowski and Jaitly, 2016) using various mechanisms, here we propose more rigorous constraints by using CTC-based alignment to guide the decoding.

CTC permits an efficient computation of a strictly monotonic alignment using dynamic programming (Graves et al., 2006; Graves and Jaitly, 2014) although it requires language models and graph-based decoding (Miao et al., 2015) except in the case of huge training data (Amodei et al., 2015; Soltau et al., 2016). We propose to take advantage of the constrained CTC alignment in a hybrid CTC/attention based system during *decoding*. The proposed method adopts a CTC/attention hybrid architecture, which was originally designed to regularize an attention-based encoder network by additionally using a CTC during *training* (Kim et al., 2017). The proposed method extends the architecture to perform one-pass/rescoring joint decoding, where hypotheses of attention-based ASR are boosted by scores obtained by using CTC outputs. This greatly reduces irregular alignments without any heuristic search techniques.

The proposed method is applied to Japanese and Mandarin ASR tasks, which require extra linguistic resources including morphological analyzer (Kudo et al., 2004) or word segmentation (Xue et al., 2003) in addition to pronunciation dictionary to provide accurate lexicon and language models in conventional DNN/HMM ASR. Surprisingly, the method achieved performance comparable to, and in some cases superior to, several state-of-the-art DNN/HMM ASR systems, without using the above linguistic resources.

2 From DNN/HMM to end-to-end ASR

This section briefly provides a formulation of conventional DNN/HMM ASR and CTC or attention based end-to-end ASR.

2.1 Conventional DNN/HMM ASR

ASR deals with a sequence mapping from T-length speech feature sequence $X = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$ to N-length word sequence $W = \{w_n \in \mathcal{V} | n = 1, \cdots, N\}$. $\mathbf{x}_t$ is a D dimensional speech feature vector (e.g., log Mel filterbanks) at frame t and w_n is a word at position n in vocabulary $\mathcal{V}$. ASR is mathematically formulated with the Bayes decision theory, where the most probable word sequence $\hat{W}$ is estimated among all possible word sequences $\mathcal{V}*$ as follows:

$$\hat{W} = \arg\max_{W \in \mathcal{V}*} p(W|X). \quad (1)$$

The posterior distribution $p(W|X)$ is factorized into the following three distributions by using the Bayes theorem and introducing HMM state sequence $S = \{s_t \in \{1, \cdots, J\} | t = 1, \cdots, T\}$:

$$\text{Eq. (1)} \approx \arg\max_{W} \sum_{S} p(X|S)p(S|W)p(W).$$

The three factors, $p(X|S)$, $p(S|W)$, and $p(W)$, are acoustic, lexicon, and language models, respectively. These are further factorized by using a probabilistic chain rule and conditional independence assumption as follows:

$$\begin{cases} p(X|S) \approx \prod_t \frac{p(s_t|\mathbf{x}_t)}{p(s_t)}, \\ p(S|W) \approx \prod_t p(s_t|s_{t-1}, W), \\ p(W) \quad \approx \prod_n p(w_n|w_{n-1}, \ldots, w_{n-m-1}), \end{cases}$$

where the acoustic model is replaced with the product of framewise posterior distributions $p(s_t|\mathbf{x}_t)$ computed by powerful DNN classifiers by using so-called pseudo likelihood trick (Bourlard and Morgan, 1994). $p(s_t|s_{t-1}, W)$ is represented by an HMM state transition given W, and the conversion from W to HMM states is deterministically performed by using a pronunciation dictionary through a phoneme representation. $p(w_n|w_{n-1}, \ldots, w_{n-m-1})$ is obtained based on an $(m-1)$th-order Markov assumption as a m-gram model.

These conditional independence assumptions are often regarded as too strong assumption, leading to model mis-specification. Also, to train the framewise posterior $p(s_t|\mathbf{x}_t)$, we have to provide a framewise state alignment s_t as a target, which is often provided by a GMM/HMM system. Thus, conventional DNN/HMM systems make the ASR problem formulated with Eq. (1) feasible by using factorization and conditional independence assumptions, at the cost of the problems discussed in Section 1.

2.2 Connectionist Temporal Classification (CTC)

The CTC formulation also follows from Bayes decision theory (Eq. (1)). Note that the CTC formulation uses L-length letter sequence $C = \{c_l \in \mathcal{U} | l = 1, \cdots, L\}$ with a set of distinct letters $\mathcal{U}$. Similarly to Section 2.1, by introducing framewise letter sequence with an additional "blank" ($<\mathtt{b}>$) symbol $Z = \{z_t \in \mathcal{U} \cup <\mathtt{b}> | t = 1, \cdots, T\}$, and by using the probabilistic chain rule and conditional independence assumption, the posterior distribution $p(C|X)$ is factorized as follows:

$$p(C|X) \approx \underbrace{\sum_{Z} \prod_{t} p(z_t|z_{t-1}, C) p(z_t|X)}_{\triangleq p_{\text{ctc}}(C|X)} \frac{p(C)}{p(Z)} \tag{2}$$

As a result, CTC has three distribution components similar to the DNN/HMM case, i.e., framewise posterior distribution $p(z_t|X)$, transition probability $p(z_t|z_{t-1}, C)$[1], and prior distributions of letter and hidden-state sequences, $p(C)$ and $p(Z)$, respectively. We also define the CTC objective function $p_{\text{ctc}}(C|X)$ used in the later formulation. The framewise posterior distribution $p(z_t|X)$ is conditioned on all inputs X, and it is quite natural to be modeled by using bidirectional long short-term memory (BLSTM): $p(z_t|X) = \text{Softmax}(\text{Lin}(\mathbf{h}_t))$ and $\mathbf{h}_t = \text{BLSTM}(X)$. $\text{Softmax}(\cdot)$ is a sofmax activation function, and $\text{Lin}(\cdot)$ is a linear layer to convert hidden vector $\mathbf{h}_t$ to a $(|\mathcal{U}|+1)$ dimensional vector ($+1$ means a blank symbol introduced in CTC).

Although Eq. (2) has to deal with a summation over all possible Z, it is efficiently computed by using dynamic programming (Viterbi/forward-backward algorithm) thanks to the Markov property. In summary, although CTC and DNN/HMM systems are similar to each other due to conditional independence assumptions, CTC does not require pronunciation dictionaries and omits an GMM/HMM construction step.

2.3 Attention mechanism

Compared with hybrid and CTC approaches, the attention-based approach does not make any conditional independence assumptions, and directly estimates the posterior $p(C|X)$ based on a probabilistic chain rule, as follows:

$$p(C|X) = \underbrace{\prod_{l} p(c_l|c_1, \cdots, c_{l-1}, X)}_{\triangleq p_{\text{att}}(C|X)}, \tag{3}$$

where $p_{\text{att}}(C|X)$ is an attention-based objective function. $p(c_l|c_1, \cdots, c_{l-1}, X)$ is obtained by

$$p(c_l|c_1, \cdots, c_{l-1}, X) = \text{Decoder}(\mathbf{r}_l, \mathbf{q}_{l-1}, c_{l-1})$$
$$\mathbf{h}_t = \text{Encoder}(X) \tag{4}$$
$$a_{lt} = \text{Attention}(\{a_{l-1}\}_t, \mathbf{q}_{l-1}, \mathbf{h}_t) \tag{5}$$
$$\mathbf{r}_l = \sum_t a_{lt} \mathbf{h}_t. \tag{6}$$

Eq. (4) converts input feature vectors X into a framewise hidden vector $\mathbf{h}_t$ in an encoder network based on BLSTM, i.e., $\text{Encoder}(X) \triangleq \text{BLSTM}(X)$. $\text{Attention}(\cdot)$ in Eq. (5) is based on a content-based attention mechanism with convolutional features, as described in (Chorowski et al., 2015) (see Appendix A). a_{lt} is an attention weight, and represents a soft alignment of hidden vector $\mathbf{h}_t$ for each output c_l based on the weighted summation of hidden vectors to form letter-wise hidden vector $\mathbf{r}_l$ in Eq. (6). A decoder network is another

[1] Note that in the implementation, the transition value is not normalized (i.e., not a probabilistic value) (Graves and Jaitly, 2014; Miao et al., 2015), similar to the HMM state transition implementation (Povey et al., 2011)

recurrent network conditioned on previous output c_{l-1} and hidden vector $\mathbf{q}_{l-1}$, similar to RNNLM, in addition to letter-wise hidden vector $\mathbf{r}_l$. We use $\text{Decoder}(\cdot) \triangleq \text{Softmax}(\text{Lin}(\text{LSTM}(\cdot)))$.

Attention-based ASR does not explicitly separate each module, and potentially handles the all issues pointed out in Section 1. It implicitly combines acoustic models, lexicon, and language models as encoder, attention, and decoder networks, which can be jointly trained as a single deep neural network.

Compared with DNN/HMM and CTC, which are based on a transition form from $t-1$ to t due to the Markov assumption, the attention mechanism does not maintain this constraint, and often provides irregular alignments. A major focus of this paper is to address this problem by using joint CTC/attention decoding.

3 Joint CTC/attention decoding

This section explains a hybrid CTC/attention network, which potentially utilizes both benefits of CTC and attention in ASR.

3.1 Hybrid CTC/attention architecture

Kim et al. (2017) uses a CTC objective function as an auxiliary task to train the attention model encoder within the multitask learning (MTL) framework, and this paper also uses the same architecture. Figure 1 illustrates the overall architecture of the framework, where the same BLSTM is shared with CTC and attention encoder networks, respectively). Unlike the sole attention model, the forward-backward algorithm of CTC can enforce monotonic alignment between speech and label sequences during training. That is, rather than solely depending on data-driven attention methods to estimate the desired alignments in long sequences, the forward-backward algorithm in CTC helps to speed up the process of estimating the desired alignment. The objective to be maximized is a logarithmic linear combination of the CTC and attention objectives, i.e., $p_{\text{ctc}}(C|X)$ in Eq. (2) and $p_{\text{att}}(C|X)$ in Eq. (3):

$$\mathcal{L}_{\text{MTL}} = \lambda \log p_{\text{ctc}}(C|X) + (1-\lambda) \log p_{\text{att}}(C|X), \tag{7}$$

with a tunable parameter $\lambda : 0 \leq \lambda \leq 1$.

3.2 Decoding strategies

The inference step of our joint CTC/attention-based end-to-end speech recognition is performed by label synchronous decoding with a beam search similar to conventional attention-based ASR. However, we take the CTC probabilities into account to find a hypothesis that is better aligned to the input speech, as shown in Figure 1. Hereafter, we describe the general attention-based decoding and conventional techniques to mitigate the alignment problem. Then, we propose joint decoding methods with a hybrid CTC/attention architecture.

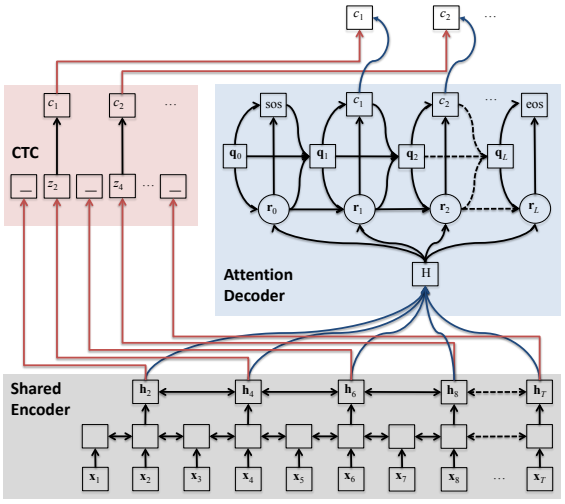


Figure 1: Joint CTC/attention based end-to-end framework: the shared encoder is trained by both CTC and attention model objectives simultaneously. The shared encoder transforms our input sequence $\{\mathbf{x}_t \cdots \mathbf{x}_T\}$ into high level features $H = \{\mathbf{h}_t \cdots \mathbf{h}_T\}$, and the attention decoder generates the letter sequence $\{c_1 \cdots c_L\}$.

3.2.1 Attention-based decoding in general

End-to-end speech recognition inference is generally defined as a problem to find the most probable letter sequence $\hat{C}$ given the speech input X, i.e.

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \log p(C|X). \tag{8}$$

In attention-based ASR, $p(C|X)$ is computed by Eq. (3), and $\hat{C}$ is found by a beam search technique.

Let Ω_l be a set of partial hypotheses of the length l. At the beginning of the beam search, Ω_0 contains only one hypothesis with the starting symbol <sos> and the hypothesis score $\alpha(\texttt{<sos>}, X)$ is set to 0. For $l = 1$ to L_{max}, each partial hypothesis in Ω_{l-1} is expanded by appending possible single letters, and the new hypotheses are stored in Ω_l, where L_{max} is the maximum

length of the hypotheses to be searched. The score of each new hypothesis is computed in the log domain as

$$\alpha(h, X) = \alpha(g, X) + \log p(c|g, X), \quad (9)$$

where g is a partial hypothesis in Ω_{l-1}, c is a letter appended to g, and h is the new hypothesis such that $h = g \cdot c$. If c is a special symbol that represents the end of a sequence, <eos>, h is added to $\hat{\Omega}$ but not Ω_l, where $\hat{\Omega}$ denotes a set of complete hypotheses. Finally, $\hat{C}$ is obtained by

$$\hat{C} = \arg\max_{h \in \hat{\Omega}} \alpha(h, X). \quad (10)$$

In the beam search process, Ω_l is allowed to hold only a limited number of hypotheses with higher scores to improve the search efficiency.

Attention-based ASR, however, may be prone to include deletion and insertion errors because of its flexible alignment property, which can attend to any portion of the encoder state sequence to predict the next label, as discussed in Section 2.3. Since attention is generated by the decoder network, it may prematurely predict the end-of-sequence label, even when it has not attended to all of the encoder frames, making the hypothesis too short. On the other hand, it may predict the next label with a high probability by attending to the same portions as those attended to before. In this case, the hypothesis becomes very long and includes repetitions of the same letter sequence.

3.2.2 Conventional decoding techniques

To alleviate the alignment problem, a length penalty term is commonly used to control the hypothesis length to be selected (Chorowski et al., 2015; Bahdanau et al., 2016). With the length penalty, the decoding objective in Eq. (8) is changed to

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\log p(C|X) + \gamma |C|\}, \quad (11)$$

where $|C|$ is the length of the sequence C, and γ is a tunable parameter. However, it is actually difficult to completely exclude hypotheses that are too long or too short even if γ is carefully tuned. It is also effective to control the hypothesis length by the minimum and maximum lengths to some extent, where the minimum and maximum are selected as fixed ratios to the length of the input speech. However, since there are exceptionally long or short transcripts compared to the input speech, it is difficult to balance saving such exceptional transcripts and preventing hypotheses with irrelevant lengths.

Another approach is the *coverage* term recently proposed in (Chorowski and Jaitly, 2016), which is incorporated in the decoding objective in Eq. (11) as

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\log p(C|X) + \gamma |C| + \eta \cdot \text{coverage}(C|X)\}, \quad (12)$$

where the coverage term is computed by

$$\text{coverage}(C|X) = \sum_{t=1}^{T} \left[\sum_{l=1}^{L} a_{lt} > \tau \right]. \quad (13)$$

η and τ are tunable parameters. The coverage term represents the number of frames that have received a cumulative attention greater than τ. Accordingly, it increases when paying close attention to some frames for the first time, but does not increase when paying attention again to the same frames. This property is effective for avoiding looping of the same label sequence within a hypothesis. However, it is still difficult to obtain a common parameter setting for γ, η, τ, and the optional min/max lengths so that they are appropriate for any speech data from different tasks.

3.2.3 Joint decoding

Our joint CTC/attention approach combines the CTC and attention-based sequence probabilities in the inference step, as well as the training step. Suppose $p_{\text{ctc}}(C|X)$ in Eq. (2) and $p_{\text{att}}(C|X)$ in Eq. (3) are the sequence probabilities given by CTC and the attention model. The decoding objective is defined similarly to Eq. (7) as

$$\hat{C} = \arg\max_{C \in \mathcal{U}^*} \{\lambda \log p_{\text{ctc}}(C|X) + (1 - \lambda) \log p_{\text{att}}(C|X)\}. \quad (14)$$

The CTC probability enforces a monotonic alignment that does not allow large jumps or looping of the same frames. Accordingly, it is possible to choose a hypothesis with a better alignment and exclude irrelevant hypotheses without relying on the coverage term, length penalty, or min/max lengths.

In the beam search process, the decoder needs to compute a score for each partial hypothesis using Eq. (9). However, it is nontrivial to combine the CTC and attention-based scores in the beam

search, because the attention decoder performs it output-label-synchronously while CTC performs it frame-synchronously. To incorporate the CTC probabilities in the hypothesis score, we propose two methods.

Rescoring

The first method is a two-pass approach, in which the first pass obtains a set of complete hypotheses using the beam search, where only the attention-based sequence probabilities are considered. The second pass rescores the complete hypotheses using the CTC and attention probabilities, where the CTC probabilities are obtained by the forward algorithm for CTC (Graves et al., 2006). The rescoring pass obtains the final result according to

$$\hat{C} = \arg\max_{h \in \hat{\Omega}} \{\lambda \alpha_{\text{ctc}}(h, X) + (1 - \lambda)\alpha_{\text{att}}(h, X)\}, \tag{15}$$

where

$$\begin{cases} \alpha_{\text{ctc}}(h, X) & \triangleq \log p_{\text{ctc}}(h|X) \\ \alpha_{\text{att}}(h, X) & \triangleq \log p_{\text{att}}(h|X) \end{cases}. \tag{16}$$

One-pass decoding

The second method is one-pass decoding, in which we compute the probability of each partial hypothesis using CTC and an attention model. Here, we utilize the CTC prefix probability (Graves, 2008) defined as the cumulative probability of all label sequences that have the partial hypothesis h as their prefix:

$$p_{\text{ctc}}(h, \ldots|X) = \sum_{\nu \in (\mathcal{U} \cup \{\texttt{<eos>}\})^+} p_{\text{ctc}}(h \cdot \nu|X),$$

and we define the CTC score as

$$\alpha_{\text{ctc}}(h, X) \triangleq \log p_{\text{ctc}}(h, \ldots|X), \tag{17}$$

where ν represents all possible label sequences except the empty string. The CTC score cannot be obtained recursively as in Eq. (9), but it can be computed efficiently by keeping the forward probabilities over the input frames for each partial hypothesis. Then it is combined with $\alpha_{\text{att}}(h, X)$.

The beam search algorithm for one-pass decoding is shown in Algorithm 1. Ω_l and $\hat{\Omega}$ are initialized in lines 2 and 3 of the algorithm, which are implemented as queues that accept partial hypotheses of the length l and complete hypotheses, respectively. In lines 4–25, each partial hypothesis g in Ω_{l-1} is extended by each label c in the label set $\mathcal{U}$. Each extended hypothesis h is scored in line 11, where CTC and attention-based scores are obtained by $\alpha_{\text{ctc}}()$ and $\alpha_{\text{att}}()$. After that, if $c =$ <eos>, the hypothesis h is assumed to be complete and stored in $\hat{\Omega}$ in line 13. If $c \neq$ <eos>, h is stored in Ω_l in line 15, where the number of hypotheses in Ω_l is checked in line 16. If the number exceeds the beam width, the hypothesis with the worst score in Ω_l is removed by REMOVEWORST() in line 17.

In line 11, the CTC and attention model scores are computed for each partial hypothesis. The attention score is easily obtained in the same manner as Eq. (9), whereas the CTC score requires a modified forward algorithm that computes it label-synchronously. The algorithm to compute the CTC score is summarized in Appendix B. By considering the attention and CTC scores during the beam search, partial hypotheses with irregular alignments can be excluded, and the number of search errors is reduced.

We can optionally apply an end detection technique to reduce the computation by stopping the beam search before l reaches L_{max}. Function ENDDETECT($\hat{\Omega}, l$) in line 22 returns `true` if there is little chance of finding complete hypotheses with higher scores as l increases in the future.

Algorithm 1 Joint CTC/attention one-pass decoding

```
1: procedure OnePassBeamSearch(X, L_max)
2:     Ω_0 ← {<sos>}
3:     Ω̂ ← ∅
4:     for l = 1 ... L_max do
5:         Ω_l ← ∅
6:         while Ω_{l-1} ≠ ∅ do
7:             g ← Head(Ω_{l-1})
8:             Dequeue(Ω_{l-1})
9:             for each c ∈ U ∪ {<eos>} do
10:                h ← g · c
11:                α(h,X) ← λα_ctc(h,X) + (1−λ)α_att(h,X)
12:                if c = <eos> then
13:                    Enqueue(Ω̂, h)
14:                else
15:                    Enqueue(Ω_l, h)
16:                    if |Ω_l| > beamWidth then
17:                        RemoveWorst(Ω_l)
18:                    end if
19:                end if
20:            end for
21:        end while
22:        if EndDetect(Ω̂, l) = true then
23:            break                          ▷ exit for loop
24:        end if
25:    end for
26:    return arg max_{h∈Ω̂} α(h, X)
27: end procedure
```

In our implementation, the function returns `true` if

$$\sum_{m=0}^{M-1}\left[\max_{h\in\hat{\Omega}:|h|=l-m}\alpha(h,X)-\max_{h'\in\hat{\Omega}}\alpha(h',X)<D_{\text{end}}\right]=M, \quad (18)$$

where D_{end} and M are predetermined thresholds. This equation becomes true if complete hypotheses with smaller scores are generated M times consecutively. This technique is also available in attention-based decoding and rescoring methods described in Sections 3.2.1–3.2.3.

4 Experiments

We used Japanese and Mandarin Chinese ASR benchmarks to show the effectiveness of the proposed joint CTC/attention decoding approach. The main reason for choosing these two languages is that those ideogram languages have relatively shorter lengths for letter sequences than those in alphabet languages, which reduces computational complexities greatly, and makes it easy to handle context information in a decoder network. Our preliminary investigation shows that Japanese and Mandarin Chinese end-to-end ASR can be easily scaled up, and shows state-of-the-art performance without using various tricks developed in English tasks. Also, we would like to emphasize that the system did not use language-specific processing (e.g., morphological analyzer, Pinyin dictionary), and simply used all appeared characters in their transcriptions including Japanese syllable and Kanji, Chinese, Arabic number, and alphabet characters, as they are.

4.1 Corpus of Spontaneous Japanese (CSJ)

We demonstrated ASR experiments by using the Corpus of Spontaneous Japanese (CSJ) (Maekawa et al., 2000). CSJ is a standard Japanese ASR task based on a collection of monologue speech data including academic lectures and simulated presentations. It has a total of 581 hours of training data and three types of evaluation data, where each evaluation task consists of 10 lectures (totally 5 hours). As input features, we used 40 mel-scale filterbank coefficients, with their first and second order temporal derivatives to obtain a total of 120-dimensional feature vector per frame. The encoder was a 4-layer BLSTM with 320 cells in each layer and direction, and linear projection layer is followed by each BLSTM layer. The 2nd and 3rd bottom layers of the encoder read every second hidden state in the network below, reducing the utterance length by the factor of 4. We used the content-based attention mechanism (Chorowski et al., 2015), where the 10 centered convolution filters of width 100 were used to extract the convolutional features. The decoder network was a 1-layer LSTM with 320 cells. The AdaDelta algorithm (Zeiler, 2012) with gradient clipping (Pascanu et al., 2012) was used for the optimization. D_{end} and M in Eq (18) were set as $\log 1e^{-10}$ and 3, respectively. The hybrid CTC/attention ASR was implemented by using the Chainer deep learning toolkit (Tokui et al., 2015).

Table 1 first compares the character error rate (CER) for conventional attention and MTL based end-to-end ASR without the joint decoding. λ in Eq. (7) was set to 0.1. When decoding, we manually set the minimum and maximum lengths of output sequences by 0.025 and 0.15 times input sequence lengths, respectively. The length penalty γ in Eq. (11) was set to 0.1. Multitask learning (MTL) significantly outperformed attention-based ASR in the all evaluation tasks, which confirms the effectiveness of a hybrid CTC/attention architecture. Table 1 also shows that joint decoding, described in Section 3.2, further improved the performance without setting any search parameters (maximum and minimum lengths, length penalty), but only setting a weight parameter $\lambda = 0.1$ in Eq. (15) similar to the MTL case. Figure 2 also compares the dependency of λ on the CER for the CSJ evaluation tasks, and showing that λ was not so sensitive to the performance if we set λ around the value we used at MTL (i.e., 0.1).

We also compare the performance of the proposed MTL-large, which has a larger network (5-layer encoder network), with the conventional state-of-the-art techniques obtained by using linguistic resources. The state-of-the-art CERs of GMM discriminative training and DNN-sMBR/HMM systems are obtained from the Kaldi recipe (Moriya et al., 2015) and a system based on syllable-based CTC with MAP decoding (Kanda et al., 2016). The Kaldi recipe systems use academic lectures (236h) for AM training and all training-data transcriptions for LM training. Unlike the proposed method, these methods use linguistic resources including a morphological analyzer, pronunciation dictionary, and language model. Note that since the amount of training

Table 1: Character error rate (CER) for conventional attention and hybrid CTC/attention end-to-end ASR. Corpus of Spontaneous Japanese speech recognition (CSJ) task.

Model	Hour	Task1	Task2	Task3
Attention	581	11.4	7.9	9.0
MTL	581	10.5	7.6	8.3
MTL + joint decoding (rescoring)	581	10.1	7.1	7.8
MTL + joint decoding (one pass)	581	10.0	7.1	7.6
MTL-large + joint decoding (rescoring)	581	**8.4**	6.2	**6.9**
MTL-large + joint decoding (one pass)	581	**8.4**	**6.1**	**6.9**
GMM-discr. (Moriya et al., 2015)	236 for AM, 581 for LM	11.2	9.2	12.1
DNN/HMM (Moriya et al., 2015)	236 for AM, 581 for LM	9.0	7.2	9.6
CTC-syllable (Kanda et al., 2016)	581	9.4	7.3	7.5

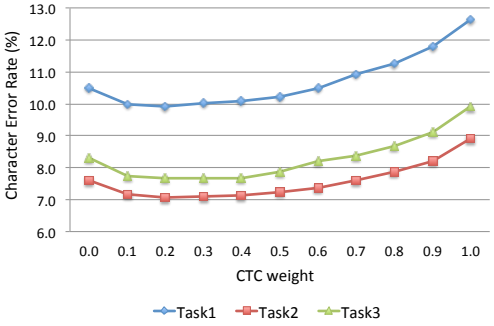


Figure 2: The effect of weight parameter λ in Eq. (14) on the CSJ evaluation tasks (The CERs were obtained by one-pass decoding).

data and experimental configurations of the proposed and reference methods are different, it is difficult to compare the performance listed in the table directly. However, since the CERs of the proposed method are superior to those of the best reference results, we can state that the proposed method achieves the state-of-the-art performance.

4.2 Mandarin telephone speech

We demonstrated ASR experiments on HKUST Mandarin Chinese conversational telephone speech recognition (MTS) (Liu et al., 2006). It has 5 hours recording for evaluation, and we extracted 5 hours from training data as a development set, and used the rest (167 hours) as a training set. All experimental conditions were same as those in Section 4.1 except that we used the $\lambda = 0.5$ in training and decoding instead of 0.1 based on our preliminary investigation and 80 mel-scale filterbank coefficients with pitch features as suggested in (Miao et al., 2016). In decoding, we also added a result of the coverage-term based decoding (Chorowski and Jaitly, 2016), as discussed in Section 3.2 ($\eta = 1.5, \tau = 0.5, \gamma = -0.6$ for attention model and $\eta = 1.0, \tau = 0.5, \gamma = -0.1$ for MTL), since it was difficult to eliminate the irregular alignments during decoding by only tuning the maximum and minimum lengths and length penalty (we set the minimum and maximum lengths of output sequences by 0.0 and 0.1 times input sequence lengths, respectively and set $\gamma = 0.6$ in Table 2).

Table 2 shows the effectiveness of MTL and joint decoding over the attention-based approach, especially showing the significant improvement of the joint CTC/attention decoding. Similar to the CSJ experiments in Section 4.1, we did not use the length-penalty term or the coverage term in joint decoding. This is an advantage of joint decoding over conventional approaches that require many tuning parameters. We also generated more training data by linearly scaling the audio lengths by factors of 0.9 and 1.1 (speed perturb.). The final model achieved **29.9**% without using linguistic resources, which defeats moderate state-of-the-art systems including CTC-based methods[2].

4.3 Decoding speed

We evaluated the speed of the joint decoding methods described in Section 3.2.3. ASR decoding was performed with different beam widths of 1, 3, 5, 10, and 20, and the processing time and CER were measured using a computer with Intel(R) Xeon(R) processors, E5-2690 v3, 2.6 GHz. Although the processors were multicore CPUs and the computer had GPUs, we ran the decoding program as a

[2] Although the proposed method did not reach the performance obtained by a time delayed neural network (TDNN) with lattice-free sequence discriminative training (Povey et al., 2016), our recent work scored **28.0**%, and outperformed the lattice-free MMI result with advanced network architectures.

Table 2: Character error rate (CER) for conventional attention and hybrid CTC/attention end-to-end ASR. HKUST Mandarin Chinese conversational telephone speech recognition (MTS) task.

Model	dev	eval
Attention	40.3	37.8
MTL	38.7	36.6
Attention + coverage	39.4	37.6
MTL + coverage	36.9	35.3
MTL + joint decoding (rescoring)	35.9	34.2
MTL + joint decoding (one pass)	35.5	33.9
MTL-large (speed perturb.) + joint decoding (rescoring)	31.1	30.1
MTL-large (speed perturb.) + joint decoding (one pass)	**31.0**	**29.9**
DNN/HMM	–	35.9
LSTM/HMM (speed perturb.)	–	33.5
CTC with language model (Miao et al., 2016)	–	34.8
TDNN/HMM, lattice-free MMI (speed perturb.) (Povey et al., 2016)	–	28.2

single-threaded process on a CPU to investigate its basic computational cost.

Table 3: RTF versus CER for the one-pass and rescoring methods.

Task	Beam width	Rescoring		One pass	
		RTF	CER	RTF	CER
CSJ Task1	1	0.66	10.9	0.66	10.7
	3	1.11	10.3	1.02	10.1
	5	1.50	10.2	1.31	10.0
	10	2.46	10.1	2.07	10.0
	20	5.02	10.1	3.76	10.0
HKUST Eval set	1	0.68	37.1	0.65	35.9
	3	0.89	34.9	0.86	34.4
	5	1.04	34.6	1.03	34.2
	10	1.55	34.4	1.50	34.0
	20	2.66	34.2	2.55	33.9

Table 3 shows the relationships between the real-time factor (RTF) and the CER for the CSJ and HKUST tasks. We evaluated the rescoring and one-pass decoding methods when using the end detection in Eq. (18). In every beam width, we can see that the one-pass method runs faster with an equal or lower CER than the rescoring method. This result demonstrates that the one-pass decoding is effective for reducing search errors. Finally, we achieved 1xRT with one-pass decoding when using a beam width around 3 to 5, even though it was a single-threaded process on a CPU. However, the decoding process has not yet achieved real-time ASR since CTC and the attention mechanism need to access all of the frames of the input utterance even when predicting the first label. This is an essential problem of most end-to-end ASR approaches and will be solved in future work.

5 Summary and discussion

This paper proposes end-to-end ASR by using joint CTC/attention decoding, which outperformed ordinary attention-based end-to-end ASR by solving the misalignment issues. The joint decoding methods actually reduced most of the irregular alignments, which can be confirmed from the examples of recognition errors and alignment plots shown in Appendix C.

The proposed end-to-end ASR does not require linguistic resources, such as morphological analyzer, pronunciation dictionary, and language model, which are essential components of conventional Japanese and Mandarin Chinese ASR systems. Nevertheless, the method achieved comparable/superior performance to the state-of-the-art conventional systems for the CSJ and MTS tasks. In addition, the proposed method does not require GMM/HMM construction for initial alignments, DNN pre-training, lattice generation for sequence discriminative training, complex search in decoding (e.g., FST decoder or lexical tree search based decoder). Thus, the method greatly simplifies the ASR building process, reducing code size and complexity.

Future work will apply this technique to the other languages including English, where we have to solve an issue of long sequence lengths, which requires heavy computation cost and makes it difficult to train a decoder network. Actually, neural machine translation handles this issue by using a sub word unit (concatenating several letters to form a new sub word unit) (Wu et al., 2016), which would be a promising direction for end-to-end ASR.

References

Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, et al. 2015. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint arXiv:1512.02595* .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 4945–4949.

Steven Bird. 2006. NLTK: the natural language toolkit. In *Joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL) on Interactive presentation sessions*. pages 69–72.

Hervé Bourlard and Nelson Morgan. 1994. *Connectionist speech recognition: A hybrid approach*. Kluwer Academic Publishers.

William Chan, Navdeep Jaitly, Quoc V Le, and Oriol Vinyals. 2015. Listen, attend and spell. *arXiv preprint arXiv:1508.01211* .

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent NN: First results. *arXiv preprint arXiv:1412.1602* .

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695* .

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems (NIPS)*. pages 577–585.

Alex Graves. 2008. Supervised sequence labelling with recurrent neural networks. *PhD thesis, Technische Universität München* .

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *International Conference on Machine learning (ICML)*. pages 369–376.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning (ICML)*. pages 1764–1772.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29(6):82–97.

Frederick Jelinek. 1976. Continuous speech recognition by statistical methods. *Proceedings of the IEEE* 64(4):532–556.

Naoyuki Kanda, Xugang Lu, and Hisashi Kawai. 2016. Maximum a posteriori based decoding for CTC acoustic models. In *Interspeech 2016*. pages 1868–1872.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 4835–4839.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Conference on Empirical Methods on Natural Language Processing (EMNLP)*. volume 4, pages 230–237.

Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff. 2006. HKUST/MTS: A very large scale mandarin telephone speech corpus. In *Chinese Spoken Language Processing*, Springer, pages 724–735.

Liang Lu, Xingxing Zhang, and Steve Renals. 2016. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 5060–5064.

Kikuo Maekawa, Hanae Koiso, Sadaoki Furui, and Hitoshi Isahara. 2000. Spontaneous speech corpus of japanese. In *International Conference on Language Resources and Evaluation (LREC)*. volume 2, pages 947–952.

Yajie Miao, Mohammad Gowayyed, and Florian Metze. 2015. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. pages 167–174.

Yajie Miao, Mohammad Gowayyed, Xingyu Na, Tom Ko, Florian Metze, and Alexander Waibel. 2016. An empirical exploration of ctc acoustic models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pages 2623–2627.

Takafumi Moriya, Takahiro Shinozaki, and Shinji Watanabe. 2015. Kaldi recipe for Japanese spontaneous speech recognition and its evaluation. In *Autumn Meeting of ASJ*. 3-Q-7.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063* .

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahrmani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. 2016. Purely sequence-trained neural networks for asr based on lattice-free MMI. In *Interspeech*. pages 2751–2755.

Hagen Soltau, Hank Liao, and Hasim Sak. 2016. Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975* .

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in NIPS*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* .

Nianwen Xue et al. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing* 8(1):29–48.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* .

A Location-based attention mechanism

This section provides the equations of a location-based attention mechanism Attention$(\cdot)$ in Eq. (5).

$$a_{lt} = \text{Attention}(\{a_{l-1}\}_t, \mathbf{q}_{l-1}, \mathbf{h}_t),$$

where $\{a_{l-1}\}_t = [a_{l-1,1}, \cdots, a_{l-1,T}]^\top$. To obtain a_{lt}, we use the following equations:

$$\{\mathbf{f}_t\}_t = \mathbf{K} * \mathbf{a}_{l-1} \tag{19}$$

$$e_{lt} = \mathbf{g}^\top \tanh(\mathbf{G}^{\text{q}}\mathbf{q}_{l-1} + \mathbf{G}^{\text{h}}\mathbf{h}_t + \mathbf{G}^{\text{f}}\mathbf{f}_t + \mathbf{b}) \tag{20}$$

$$a_{lt} = \frac{\exp(e_{tl})}{\sum_t \exp(e_{tl})} \tag{21}$$

$\mathbf{K}$, $\mathbf{G}^{\text{q}}$, $\mathbf{G}^{\text{h}}$, $\mathbf{G}^{\text{f}}$ are matrix parameters. $\mathbf{b}$ and $\mathbf{g}$ are vector parameters. $*$ denotes convolution along input feature axis t with matrix $\mathbf{K}$ to produce feature $\{\mathbf{f}_t\}_t$.

Algorithm 2 CTC hypothesis score

1: **function** $\alpha_{\text{ctc}}(h, X)$
2: $g, c \leftarrow h$ ▷ split h into the last label c and the rest g
3: **if** $c =$ <eos> **then**
4: **return** $\log\{\gamma_T^{(n)}(g) + \gamma_T^{(b)}(g)\}$
5: **else**
6: $\gamma_1^{(n)}(h) \leftarrow \begin{cases} p(z_1 = c|X) & \text{if } g = \text{<sos>} \\ 0 & \text{otherwise} \end{cases}$
7: $\gamma_1^{(b)}(h) \leftarrow 0$
8: $\Psi \leftarrow \gamma_1^{(n)}(h)$
9: **for** $t = 2 \dots T$ **do**
10: $\Phi \leftarrow \gamma_{t-1}^{(b)}(g) + \begin{cases} 0 & \text{if last}(g) = c \\ \gamma_{t-1}^{(n)}(g) & \text{otherwise} \end{cases}$
11: $\gamma_t^{(n)}(h) \leftarrow \left(\gamma_{t-1}^{(n)}(h) + \Phi\right) p(z_t = c|X)$
12: $\gamma_t^{(b)}(h) \leftarrow \left(\gamma_{t-1}^{(b)}(h) + \gamma_{t-1}^{(n)}(h)\right) p(z_t =$ <b>$|X)$
13: $\Psi \leftarrow \Psi + \Phi \cdot p(z_t = c|X)$
14: **end for**
15: **return** $\log(\Psi)$
16: **end if**
17: **end function**

B CTC-based hypothesis score

The CTC score $\alpha_{\text{ctc}}(h, X)$ in Eq. (17) is computed as shown in Algorithm 2. Let $\gamma_t^{(n)}(h)$ and $\gamma_t^{(b)}(h)$ be the forward probabilities of the hypothesis h over the time frames $1 \dots t$, where the superscripts (n) and (b) denote different cases in which all CTC paths end with a nonblank or blank symbol, respectively. Before starting the beam search, $\gamma_t^{(n)}()$ and $\gamma_t^{(b)}()$ are initialized for $t = 1, \dots, T$ as

$$\gamma_t^{(n)}(\text{<sos>}) = 0, \tag{22}$$

$$\gamma_t^{(b)}(\text{<sos>}) = \prod_{\tau=1}^{t} \gamma_{\tau-1}^{(b)}(\text{<sos>}) p(z_\tau = \text{<b>}|X), \tag{23}$$

where we assume that $\gamma_0^{(b)}(\text{<sos>}) = 1$ and <b> is a blank symbol. Note that the time index t and input length T may differ from those of the input utterance X owing to the subsampling technique for the encoder (Povey et al., 2016; Chan et al., 2015).

In Algorithm 2, the hypothesis h is first split into the last label c and the rest g in line 2. If c is <eos>, it returns the logarithm of the forward probability assuming that h is a complete hypothesis in line 4. The forward probability of h is given by

$$p_{\text{ctc}}(h|X) = \gamma_T^{(n)}(g) + \gamma_T^{(b)}(g) \tag{24}$$

according to the definition of $\gamma_t^{(n)}()$ and $\gamma_t^{(b)}()$. If c is not <eos>, it computes the forward proba-

bilities $\gamma_t^{(n)}(h)$ and $\gamma_t^{(b)}(h)$, and the prefix probability $\Psi = p_{\text{ctc}}(h, \ldots |X)$ assuming that h is not a complete hypothesis. The initialization and recursion steps for those probabilities are described in lines 6–14. In this function, we assume that whenever we compute the probabilities $\gamma_t^{(n)}(h)$, $\gamma_t^{(b)}(h)$ and Ψ, the forward probabilities $\gamma_t^{(n)}(g)$ and $\gamma_t^{(b)}(g)$ have already been obtained through the beam search process because g is a prefix of h such that $|g| < |h|$.

C Examples of irregular alignments

We list examples of irregular alignments caused by attention-based ASR. Figure 3 shows an example of repetitions of word chunks. The first chunk of blue characters in attention-based ASR (MTL) is appeared again, and the whole second chunk part becomes insertion errors. Figure 4 shows an example of deletion errors. The latter half of the sentence in attention-based ASR (MTL) is broken, which causes deletion errors. The hybrid CTC/attention with both multitask learning and joint decoding avoids these issues. Figures 5 and 6 show alignment plots corresponding to Figs. 3 and 4, respectively, where X-axis shows time frames and Y-axis shows the character sequence hypothesis. These visual plots also demonstrate that the proposed joint decoding approach can suppress irregular alignments.

id: (20040717_152947_A010409_B010408-A-057045-057837)
Reference
但 是 如 果 你 想 想 如 果 回 到 了 过 去 你 如 果 带 着 这 个 现 在 的 记
忆 是 不 是 很 痛 苦 啊

MTL
Scores: (#Correctness #Substitution #Deletion #Insertion) 28 2 3 45
但 是 如 果 你 想 想 如 果 回 到 了 过 去 你 如 果 带 着 这 个 现 在 的 节
如 果 你 想 想 如 果 回 到 了 过 去 你 如 果 带 着 这 个 现 在 的 节 如 果
你 想 想 如 果 回 到 了 过 去 你 如 果 带 着 这 个 现 在 的 机 是 不 是 很
・ ・ ・

Joint decoding
Scores: (#Correctness #Substitution #Deletion #Insertion) 31 1 1 0
HYP: 但 是 如 果 你 想 想 如 果 回 到 了 过 去 你 如 果 带 着 这 个 现
在 的 ・ 机 是 不 是 很 痛 苦 啊

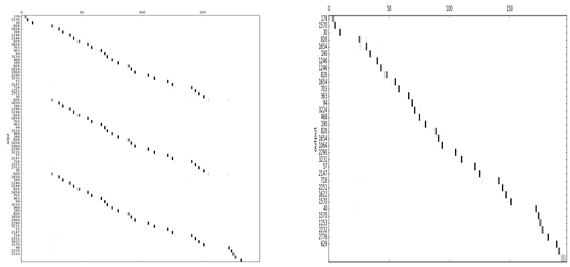
Figure 3: Example of insertion errors appeared in attention-based ASR with MTL and joint decoding.

id: (A01F0001_0844951_0854386)
Reference
ま た え 飛 行 時 の エ コ ー ロ ケ ー シ ョ ン 機 能 を よ り
詳 細 に 解 明 す る 為 に 超 小 型 マ イ ク ロ ホ ン お よ び
生 体 ア ン プ を コ ウ モ リ に 搭 載 す る こ と を 考 え て
お り ま す そ う す る こ と に よ っ て

MTL
Scores: (#Correctness #Substitution #Deletion #Insertion) 30 0 47 0
ま た え 飛 行 時 の エ コ ー ロ ケ ー シ ョ ン 機 能 を よ り
詳 細 に 解 明 す る 為 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
・ ・
・ ・ ・ ・ ・ ・ ・ ・ ・ ・ に ・ ・ ・

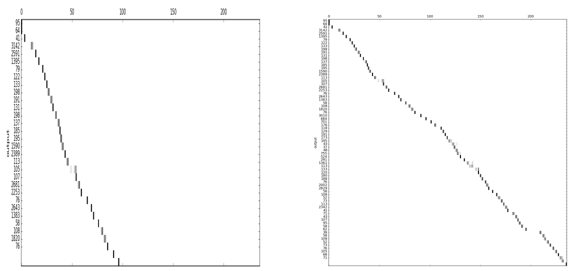
Joint decoding
Scores: (#Correctness #Substitution #Deletion #Insertion) 67 9 1 0
ま た え 飛 行 時 の エ コ ー ロ ケ ー シ ョ ン 機 能 を よ り
詳 細 に 解 明 す る 為 に 長 国 型 マ イ ク ロ ホ ン お ・ い
く 声 単 位 方 を コ ウ モ リ に 登 載 す る こ と を 考 え て
お り ま す そ う す る こ と に よ っ て

Figure 4: Example of deletion errors appeared in attention-based ASR with MTL and joint decoding.



(a) MTL (b) Joint decoding

Figure 5: Example of alignments including insertion errors in attention-based ASR with MTL and joint decoding (Utterance id: 20040717_152947_A010409_B010408-A-057045-057837).



(a) MTL (b) Joint decoding

Figure 6: Example of alignments including deletion errors in attention-based ASR with MTL and joint decoding (Utterance id: A01F0001_0844951_0854386).