Hybrid CTC-Attention based End-to-End Speech Recognition using Subword Units

Zhangyu Xiao[1], Zhijian Ou[1], Wei Chu[2], Hui Lin[2]

[1]Speech Processing and Machine Intelligence (SPMI) Lab,
Tsinghua University, Beijing, China.

[2]Liulishuo Information Technology Co., Ltd.
Shanghai, China.

November 2018

Tsinghua University 流利说

Contents

1. Background
   - Model structure
   - Modeling Units
2. Hybrid CTC-Attention end-to-end ASR
   - CTC & Attention
   - Experiment Results
3. Subword Units
   - Definition & example
   - Experiment Results

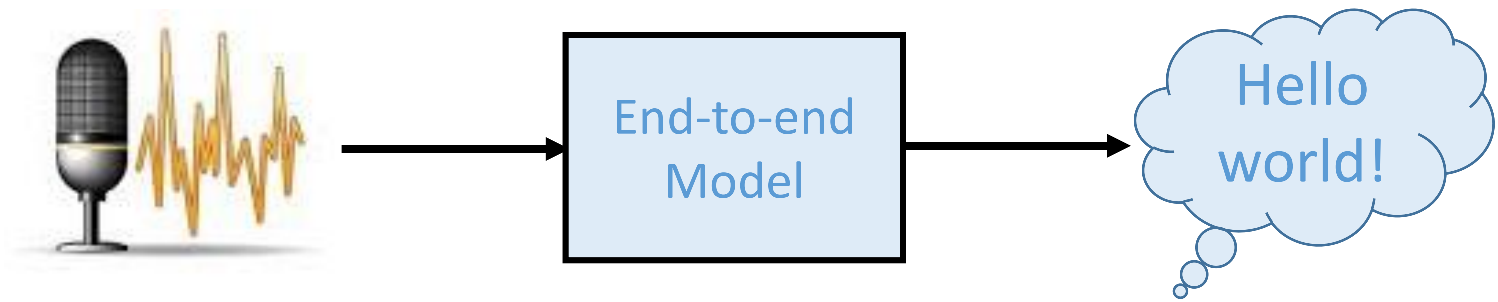
Contents

1. Background
    - Model structure
    - Modeling Units
2. Hybrid CTC-Attention end-to-end ASR
3. Subword Units

Background

End-to-end Speech Recognition

- a single system that directly transcribes speech signal to words
- usually based on NN structures and can be trained from scratch



Background——Model Structure

End-to-end Speech Recognition

- a single system that directly transcribes speech signal to words
- usually based on NN structures and can be trained from scratch

CTC based

- makes a strong independent assumption between labels
- estimates alignment with forward-backward algorithm

Attention based

- attention decoder emits labels depending on previous ones
- hard to train due to its excessively flexible attention alignments

Background——modeling units

End-to-end Speech Recognition

- a single system that directly transcribes speech signal to words
- usually based on NN structures and can be trained from scratch

- phonemes “AA,AE,…”
- characters “a,b,c,d,…”
- subwords “abs,ing,…”
- words “hello,hi,…”

Contents

1. Background
2. Hybrid CTC-Attention end-to-end ASR
   - CTC & Attention
   - Experiment Results
3. Subword Units

Hybrid CTC-Attention end-to-end ASR

CTC based model:

- makes a strong independent assumption between labels

$$p(y|x) = \sum_{\pi \in \phi(y)} p(\pi|x) = \sum_{\pi \in \phi(y)} \prod_{l=1}^{L} q_l^{\pi_l}$$

☹ can not perform well without language model

- estimates alignment with forward-backward algorithm

☺ easy to train and converge

A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks", ICML, 2006.

Hybrid CTC-Attention end-to-end ASR

Attention based model:

- the attention decoder emits labels depending on previous ones

$$p(y|x) = \prod_{u} p(y_u|h, y_{1:u})$$

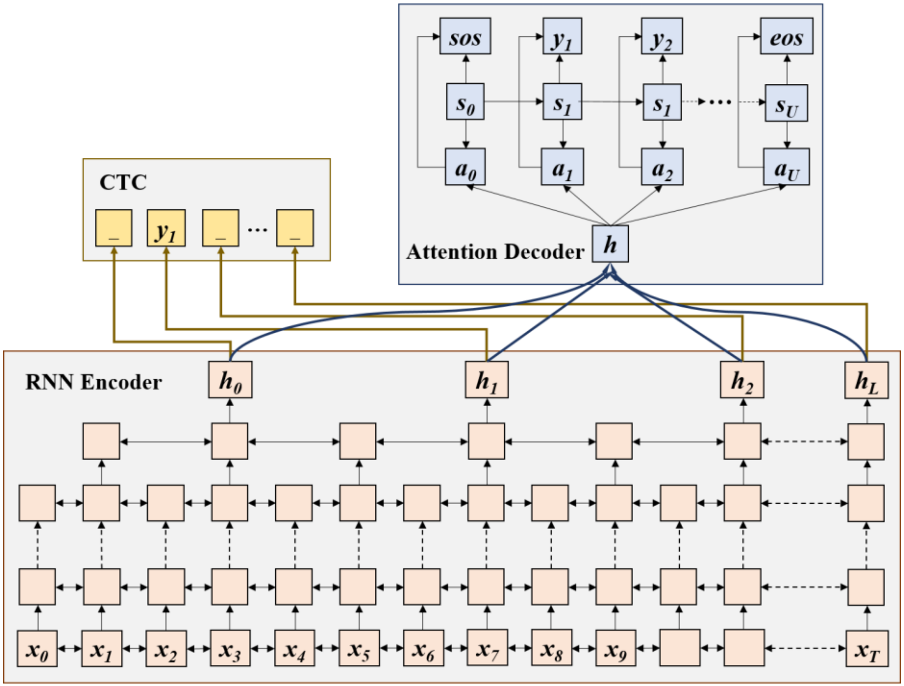
☺ can model label dependencies

- excessively flexible attention alignments

☹ hard to train and converge

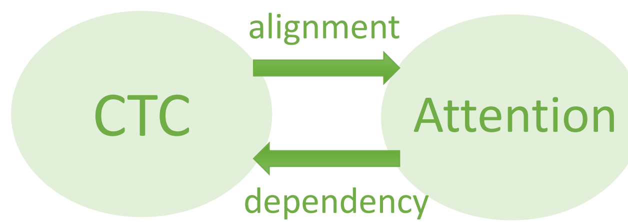
D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," ICASSP, 2016.

Hybrid CTC-Attention end-to-end ASR



Hybrid CTC-Attention:

- Pyramidal BLSTM based RNN Encoder
- CTC and Attention Decoder share the same RNN Encoder



$$L_{hybrid} = \lambda L_{CTC} + (1 - \lambda) L_{Att}$$

S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based endto-end speech recognition using multi-task learning," ICASSP, 2017.

Hybrid CTC-Attention end-to-end ASR

Table: Results of different e2e model structures on Librispeech

Model	λ	Word Error Rate/%			
		test-clean	test-other	dev-clean	dev-other
CTC	1.0	20.9	39.8	21.4	38.6
Attention	0.0	10.5	30.9	9.9	28.6
CTC+Attention	0.2	**7.8**	**21.9**	**7.7**	**21.3**

- different λ determine different model structures
- CTC cannot perform well without a LM
- The hybrid CTC-Attention model outperforms both of CTC and Attention based models!

Contents

1. Background

2. Hybrid CTC-Attention end-to-end ASR

3. Subword units

   - Definition & example

   - Experiment Results

Subword Units

Table: Examples of different modeling units

Basic Units	Segmented Sequence
word	that neither of them had crossed the threshold since the dark day
phoneme	DH AE1 T N IY1 DH ER0 AH1 V DH EH1 M HH AE1 D K R AO1 S T DH AH0 TH R EH1 SH OW2 L D S IH1 N S DH AH0 D AA1 R K D EY1
character	t h a t _ n e i t h e r _ o f _ t h e m _ h a d _ c r o s s e d _ t h e _ t h r e s h o l d _ s i n c e _ t h e _ d a r k _ d a y _
subword	that_ ne i ther_ of_ them_ had_ cro s sed_ the_ th re sh old_ sin ce_ the_ d ar k_ day_

- phoneme based on CMUDICT
- special symbol "_" denotes word boundary

Subword Units

Table: Comparison of different modeling units

Basic Units	Total Number	Length of sequence	Ability of handling OOV
word	$N * 10^{4\sim5}$	shortest/12	NO
phoneme	$N * 10$	Long/41	NO
character	$N * 10$	Longest/66	YES
subword	$N * 10^{2\sim3}$	Short/22	YES

Numbers in length of sequence:

takes the utterance of the former page as example

Large total number

- heavy computation cost due to softmax
- label spareness

Long output seq

- difficult to capture word-level dependency
- easy to generate substitution error

Fixed dictionay

- unable to handle the Out-Of-Vocabulary problem

Subword Units

Table: Comparison of different modeling units

Basic Units	Total Number	Length of sequence	Ability of handling OOV
word	$N * 10^{4\sim5}$	shortest/12	NO
phoneme	$N * 10$	Long/41	NO
character	$N * 10$	Longest/66	YES
subword	$N * 10^{2\sim3}$	Short/22	YES

Numbers in length of sequence:

takes the utterance of the former page as example

Large total number

- heavy computation cost due to softmax
- label sparseness

Long output seq

- difficult to capture word-level dependency
- easy to generate substitution error

Fixed dictionay

- unable to handle the Out-Of-Vocabulary problem

Subword Units: Generation & Segmentation

Subword Generation Algorithm: Byte-Pair Encoding(BPE)

BPE Algorithm	
Step 1.	Initialize subword set S with 26 charaters and word boundary symbol "_": $S = \{a,b,c,\ldots,z,\_\}$
Step 2.	Count all symbol pairs, and find the most frequent pair (c^1,c^2)
Step 3.	Merge the most frequent pair to a new symbol "c^1c^2", and add it to S
Step 4.	If $\|S\| < N$ (a predefined number), go to Step 2. Else, go to Step 5.
Step 5.	Output the final subword set S of size N.

Subword Units: Experiment Results

Table: Experiments on Librispeech 1000h Dataset

Model	Basic unit	λ	Word Error Rate/%			
			test-clean	test-other	dev-clean	dev-other
CTC	char	1.0	20.9	39.8	21.4	38.6
Att	char	0.0	10.5	30.9	9.9	28.6
CTC+Att	char	0.2	7.8	21.9	7.7	21.3
CTC+Att	subword	0.2	**6.8**	**19.5**	**6.7**	**18.8**

Basic Unit	WER	Sub	Del	Ins
char	7.8	**6.4**	0.6	0.8
subword	6.8	**5.4**	0.5	0.9

- significant improvement from character to subword:
  - relatively 12.8% WER reduction
  - Mostly from substitution error

Subword Units: Experiment Results

Figure 1: Influence of λ

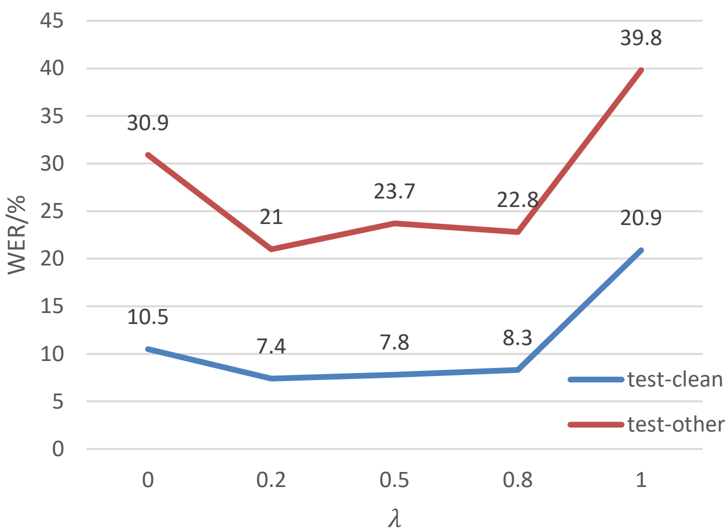
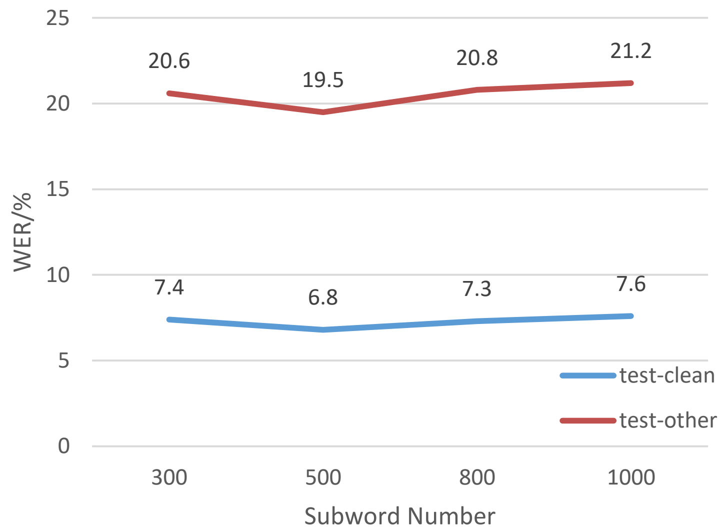


Figure 2: Influence of subword number



- CTC should form a small proportion in the hybrid loss

- Number of subword units should not be too large nor too small.

Thank you!

Any Questions?