Subword Regularization and Beam Search Decoding for End-to-End Automatic Speech Recognition

Jennifer Drexler   James Glass

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139

Overview

Motivation

- Extend subword regularization (Kudo 2018) from machine translation to ASR
- Apply subword regularization to both attention-based and CTC-based ASR models
- Understand interactions between subword regularization and use of language model during decoding

Innovations

- ASR-specific modifications to subword unit discovery procedure
- Developed, implemented, and released (https://github.com/jdrex/ctcdecode) subword prefix beam search decoding algorithm for CTC

Results

- Subword regularization improves ASR performance in all cases, is especially effective with attention-based models
- Novel subword prefix beam search decoding algorithm is necessary for use of subword regularization with CTC-based models
- Improvements from subword regularization are complementary with language model addition

Baseline Models and Data

- Listen, Attend, and Spell architecture (Chan, 2016) for attention-based ASR
- Variant of Deepspeech2 architecture (Amodei, 2016) for CTC-based ASR
- Data: Wall Street Journal (WSJ) and Librispeech corpora
  - Standard train/dev/test splits
- Word-level 4-gram language models
  - Included in beam search with WFST composition

Subword Regularization

- Jointly learn vocabulary of subword units and a probabilistic model for segmenting text
  - Enables use of different segmentation of target text on each training iteration
  - Produces large gains over BPE when used with high-quality attention-based machine translation models
- Unit discovery procedure
  - Initialize with very large vocabulary of most common subword units in text corpus
  - Train unigram language model
  - Remove 5% of units that contribute least to data likelihood
  - Iterate over training procedure until desired vocabulary size is reached
- Segmentation procedure
  - Single best segmentation (or n-best list) can be found with Vitterbi search
  - Segmentations can be sampled from the following multinomial distribution:

$$P(x_i|X) \cong \frac{p(x_i)^\alpha}{\sum_{j=1}^{n} p(x_j)^\alpha}$$

  - n is the number of n-best segmentations used to approximate the true distribution
  - α is the regularization parameter: $\alpha = 0$ creates a uniform distribution, increasing α moves closer to the Viterbi segmentation

Modifications for ASR

- Our goal: capture acoustic/phonetic properties, not semantics
  - Limit length (in characters) of discovered units
  - Small vocabulary
  - Spaces are always a separate, single character
- Example Segmentations (WSJ)

$\lvert V\rvert$	maxlen	method	segmentation
5000	∞	best	`HISTORICAL LY`
500	4	best	`HIS T OR ICAL LY`
		sample	`HIS TO RI CALL Y`
		sample	`H IS TO R ICAL LY`

Results - Attention

Segmentation	α	WER No LM	WER + LM
Character		16.0	12.4
Unigram, 100 units, ≤ 2	∞	16.0	12.1
	1	14.1	**10.7**
	0.5	14.2	11.6
	0.2	14.3	11.5
Unigram, 200 units, ≤ 4	∞	15.1	11.8
	1	**14.0**	**10.7**
	0.5	14.3	11.1
	0.2	14.8	11.0

Table: Results from the encoder-decoder model with attention on the WSJ dataset.

Subword Beam Search for CTC

- Prefix Beam Search Decoding
  - Keep n prefixes with highest cumulative probability at time t:

$$p(\boldsymbol{p}|x,t) = \gamma(\boldsymbol{p}_b,t) + \gamma(\boldsymbol{p}_n,t)$$

  - $\gamma(\boldsymbol{p}_b,t)$ is the probability of outputting prefix $\boldsymbol{p}$ by time t such that the blank label is output at time t
  - $\gamma(\boldsymbol{p}_n,t)$ is the probability of outputting prefix $\boldsymbol{p}$ by time t such that a non-blank label is output at time t
- Problem: same prefix can be generated with different sequences of subword units
  - Valid outputs for prefix **CAT**: **C—A—T**, **CA—T**, **C—AT**, **C—AT—AT**, **CAT**
  - Standard algorithm would assign these 5 options to 4 different prefixes
  - Simplest solution (check match of overall character string) would collapse all of the above plus these invalid outputs: **CA—A—T**, **CAT—T**
- Subword Prefix Beam Search Decoding
  - Maximum subword unit length M
  - Updated prefix probability:

$$p(\boldsymbol{p}|x,t) = \gamma(\boldsymbol{p}_b,t) + \sum_{z=1}^{M} \gamma(\boldsymbol{p}_n,z,t)$$

  - $\gamma(\boldsymbol{p}_n,z,t)$ is the probability of outputting prefix $\boldsymbol{p}$ by time t such that a non-blank label **of length z** is output at time t

Results - CTC

Segmentation	α	WER (no LM)	sWER (no LM)	sWER (+ LM)
Character		19.8	19.8	16.1
Unigram, 100, ≤ 2	∞	20.0	20.0	15.1
	10	19.8	19.5	14.1
	5	19.4	**18.8**	**14.0**
	2	22.0	19.5	14.8
	1	28.5	20.6	15.5
	0.5	37.9	22.0	15.7

Table: Results from the CTC model on the WSJ dataset. WER denotes results using the standard prefix beam search algorithm; sWER results use our updated algorithm.

Segmentation	α	clean sWER (+ LM)	other sWER (+ LM)
Character		11.9 (8.3)	31.1 (24.4)
Unigram, 200, ≤ 3	∞	11.9 (8.1)	30.5 (23.1)
	2	12.3 (**7.4**)	30.4 (22.0)
	1	12.4 (7.7)	30.2 (22.5)
	0.5	13.8 (8.9)	31.8 (24.6)
Unigram, 500, ≤ 4	∞	**11.7** (8.2)	29.9 (23.0)
	2	12.6 (7.8)	29.9 (**21.7**)
	1	12.1 (8.0)	**29.4** (22.4)
	0.5	12.4 (9.7)	30.7 (25.3)

Table: Results from the CTC model on the Librispeech dataset.

Conclusions

- Subword regularization is effective for ASR
  - Larger improvements with attention-based than with CTC-based model
  - CTC-based model requires modified beam search decoding for optimal performance
- More analysis needed on choice of subword vocabulary
  - Comparison with Gram-CTC (Liu, 2017)
  - Interaction with language model