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

Results - Attention

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>Character</th>
<th>α</th>
<th>No LM</th>
<th>LM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>16.0</td>
<td>12.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unigram, 100 units, ≤ 2</td>
<td>16.0</td>
<td>12.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14.1</td>
<td>10.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>14.2</td>
<td>11.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>14.3</td>
<td>11.5</td>
<td></td>
</tr>
<tr>
<td>Unigram, 200 units, ≤ 2</td>
<td>17.1</td>
<td>14.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>14.0</td>
<td>10.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>14.3</td>
<td>11.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>14.8</td>
<td>11.0</td>
<td></td>
</tr>
</tbody>
</table>

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

Results - CTC

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>Character</th>
<th>α</th>
<th>clean SWER (+ LM)</th>
<th>other SWER (+ LM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character</td>
<td>11.9</td>
<td>8.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unigram, 200 units, ≤ 3</td>
<td>11.9 (8.1)</td>
<td>30.5 (23.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12.3</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>12.4</td>
<td>7.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>13.8</td>
<td>8.9</td>
<td></td>
</tr>
<tr>
<td>Unigram, 500 units, ≤ 4</td>
<td>11.7 (8.2)</td>
<td>29.9 (23.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12.6</td>
<td>7.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>12.1 (8.0)</td>
<td>29.4 (22.4)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>12.4 (9.7)</td>
<td>30.7 (25.3)</td>
<td></td>
</tr>
</tbody>
</table>

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.

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 Segments (WSJ)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>maxlen method</td>
<td>segmentation</td>
</tr>
<tr>
<td>500</td>
<td>best</td>
<td>historical, ly</td>
</tr>
<tr>
<td>300</td>
<td>best</td>
<td>his to r ical ly</td>
</tr>
</tbody>
</table>

Subword Beam Search for CTC
- Prefix Beam Search Decoding
  - Keep n prefixes with highest cumulative probability at time t:
    \[ p(p(x, t) = \gamma(p_n, t) + \gamma(p_{n-1}, t) \]
  - \( \gamma(p_n, t) \) is the probability of outputting prefix \( p \) by time \( t \) such that a non-blank label is output at time \( t \)
  - \( \gamma(p_n, t) \) is the probability of outputting prefix \( p \) 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
  - 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:
  \[ p(x, t) = \gamma(p_n, t) \]
- \( \gamma(p_n, z, t) \) is the probability of outputting prefix \( p \) by time \( t \) such that a non-blank label of length \( z \) is output at time \( t \)

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

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 Viterbi search
  - Segments can be sampled from the following multinomial distribution:
    \[ P(x|X) = \frac{p(x|n)}{\sum_{x|n} p(x|n)} \]
    - \( n \) is the number of n-best list segmentations used to approximate the true distribution
    - \( \alpha \) is the regularization parameter: \( \alpha = 0 \) creates a uniform distribution, increasing \( \alpha \) moves closer to the Viterbi segmentation

Limitation
- \( \alpha \) is the probability of outputting prefix \( p \) by time \( t \) such that a non-blank label is output at time \( t \)
- \( \gamma(p_n, t) \) is the probability of outputting prefix \( p \) 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
  - 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:
    \[ p(x, t) = \gamma(p_n, t) \]
  - \( \gamma(p_n, z, t) \) is the probability of outputting prefix \( p \) by time \( t \) such that a non-blank label of length \( z \) is output at time \( t \)

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