Attention-based ASR utilizing Byte-Pair Encoding

Albert Zeyer, Ralf Schlüter
Human Language Technology and Pattern Recognition — Lehrstuhl Informatik 6
Fakultät für Mathematik, Informatik und Naturwissenschaften
RWTH Aachen University
End-to-end.

Current state-of-the-art: hybrid HMM/ANN approach
- usually based on initial Gaussian mixture HMM training
- operates on phone level using pronunciation lexica and full word forms
- end-to-end scheme possible:
  - lattice-free MMI [Povey+ 2016]
  - GMM-free incl. phonetic decision tree [Gosztolya+ 2016]

What is "end-to-end"?
- lower end: input features, e.g. MFCCs
- upper end: output labels, e.g. characters, words, subwords; here: byte-pair encoding (BPE)
- aim: homogeneous modeling, training and decoding
- (no pronunciation lexicon, no phone model and clustering)

Models:
- HMM, CTC, ASG, LF-MMI
- encoder-decoder with attention (here)
- inverted HMM / segmental RNN, recurrent transducer, recurrent neural aligner

Except for HMM, discriminative model: \( p(y_1^N|x_1^T) = \prod_{i=1}^N p(y_i|[y_1^{i-1}], x_1^T) \)
Model: Encoder-decoder with attention

Encoder

• high-level feature representation/transformation
• deep bi-directional LSTM network
• max-pooling in time: optional sub-sampling following each LSTM layer
• input feature vector sequence $x_1^T$,
• encoder output:

$$h_{1'}^T = \text{LSTM}_{\#\text{enc}} \circ \cdots \circ \text{max-pool}_1 \circ \text{LSTM}_1(x_1^T),$$

- $T' = \text{red} \cdot T$ with time reduction factor $\text{red}$,
- $\#\text{enc}$: number of encoder layers, $\#\text{enc} \geq 2$. 
Model: Encoder-decoder with attention.

Decoder

• attention energies for encoder time-step $t$ and decoder step (output label position) $i$:

$$e_{i,t} = \mathbf{v}^\top \tanh(W[s_i, h_t, \beta_{i,t}]),$$

with trainable vector $\mathbf{v}$ and trainable matrix $W$, current decoder state $s_i$, and encoder state $h_t$.

**Note:** no dependence on symbol $y_i$ to be hypothesized in position $i$!

• attention weight feedback: influence of attention used in earlier decoder steps

$$\beta_{i,t} = \sigma(\mathbf{v}_\beta^\top h_t) \cdot \sum_{k=1}^{i-1} \alpha_{k,t}, \text{ with trainable vector } \mathbf{v}_\beta.$$

• attention weights: $\alpha_i = \text{softmax}_t(e_i)$, normalized over time

• attention context vector: input to decoder

$$c_i = \sum_{t=1}^{T'} \alpha_{i,t} h_t$$

• decoder state: $s_i = \text{LSTMCell}(s_{i-1}, y_{i-1}, c_{i-1})$

• decoder output prediction probability:

$$p(y_i|y_{1}^{i-1}, x_T) = \text{softmax}(\text{linear} \circ \text{maxout} \circ \text{linear}(s_i, y_{i-1}, c_i))$$
Model: Encoder-decoder with attention.

Attention Example

"that's really interesting i've never heard of anybody making their own pudding before that's really neat"

- grayscale reflects attention weights for each time frame
Pretraining

Time reduction and pretraining

- input sequences are much longer than output sequences (e.g. 30 times)
- continuous input: downscaling / time reduction can work
- time reduction factors:
  - low, e.g. 8 or less: hard to get to convergence
    (did not converge at all for us in most cases, or was very bad)
  - high, e.g. 16 or 32: converges fast and nicely

- deep hybrid LSTM models can benefit from layer-wise pretraining:
  start with 1 or 2 layers, add more and more layers
- same for the deep LSTM encoder
  - first we showed that it works for machine translation
  - also works for speech recognition
- pretraining in speech recognition with time reduction scheduling:
  - start with high time reduction (32), and then reduce (to 8)
  - get better performance with lower time reduction
- pretraining with other scheduling variants:
  - label smoothing (initially disabled)
  - dropout (initially disabled)
Pretraining

- Pretraining study for machine translation (WMT 2017 German→English task):

<table>
<thead>
<tr>
<th>encoder num. layers</th>
<th>BLEU [%]</th>
<th>no pretrain</th>
<th>with pretrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>29.3</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>29.9</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>29.1</td>
<td>30.3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>30.3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>30.6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>30.9</td>
<td></td>
</tr>
</tbody>
</table>

- Time reduction for ASR (Switchboard), directly starting with:
  - time reduction factor 8, 2 layers, or
  - time reduction factor 32, 6 layers:
did not work
- might work with more careful tuning, not needed with pretraining
Experiments

Optimal time reduction factor

• always with pretraining, starting with time reduction factor 32
• Switchboard 300h, Hub5’00 (SWB+CH) results:

<table>
<thead>
<tr>
<th>factor</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(out of memory)</td>
</tr>
<tr>
<td>8</td>
<td>20.4</td>
</tr>
<tr>
<td>16</td>
<td>21.0</td>
</tr>
<tr>
<td>32</td>
<td>21.9</td>
</tr>
</tbody>
</table>

• (factors in between not straightforward with our max-pooling time reduction)
Experiments

**Byte-Pair Encoding**

**Goals**: enable open vocabulary and avoid pronunciation lexicon

- **character vocabulary**:
  - small label set
  - enables open vocabulary
  - pronunciation ambiguous/context dependent

- **word vocabulary**:
  - large label set
  - fixed/limited vocabulary
  - pronunciations well-defined

**Byte-pair encoding**:

- starts from character vocabulary and corresponding segmentation
- iteratively merges most frequent label bigrams ("byte-pairs")
- always keeps byte-pair constituents in vocabulary
- word internal processing only, word internal boundary symbol '@', attached to left character
- leads to intermediate size of label set
Byte-Pair Encoding: Beyond Zipf’s Law [Montemurro 2001]

- rank $r$ vs. frequency $N(r)$ in training corpus
Experiments

Switchboard 300h

- 6 layer encoder attention model, 1 layer decoder
- \(^1\) added noise from external data. \(^2\) added the lexicon, i.e. also additional data.

<table>
<thead>
<tr>
<th>model</th>
<th>LM</th>
<th>label unit</th>
<th>WER[%]</th>
<th>Hub5’00 SWB</th>
<th>CH</th>
</tr>
</thead>
<tbody>
<tr>
<td>JHU, LF MMI, 2016</td>
<td>4-gram</td>
<td>CDp</td>
<td></td>
<td>9.6</td>
<td>19.3</td>
</tr>
<tr>
<td>hybrid (this work)</td>
<td>LSTM</td>
<td></td>
<td></td>
<td>8.3</td>
<td>17.3</td>
</tr>
<tr>
<td>Edinburgh, attention, 2016</td>
<td>3-gram</td>
<td>words</td>
<td></td>
<td>25.8</td>
<td>46.0</td>
</tr>
<tr>
<td>Toyota, attention, 2017</td>
<td>none</td>
<td>chars</td>
<td></td>
<td>23.1</td>
<td>40.8</td>
</tr>
<tr>
<td>Stanford, CTC, 2015</td>
<td>RNN</td>
<td>chars</td>
<td></td>
<td>21.4</td>
<td>40.2</td>
</tr>
<tr>
<td>Baidu DeepSpeech, CTC(^1), 2014</td>
<td></td>
<td></td>
<td></td>
<td>20.0</td>
<td>31.8</td>
</tr>
<tr>
<td>Microsoft, CTC(^2), 2017</td>
<td>word RNN</td>
<td></td>
<td></td>
<td>14.0</td>
<td>25.3</td>
</tr>
<tr>
<td>attention (this work)</td>
<td>none</td>
<td>BPE</td>
<td></td>
<td>10K</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1K</td>
<td></td>
<td>13.1</td>
<td>26.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1K</td>
<td></td>
<td>11.8</td>
<td>25.7</td>
</tr>
</tbody>
</table>
## LibriSpeech 1000h

- 6 layer encoder attention model, 1 layer decoder

<table>
<thead>
<tr>
<th>model</th>
<th>LM</th>
<th>label unit</th>
<th>WER [%]</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>dev</td>
<td>test</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clean</td>
<td>other</td>
<td>clean</td>
</tr>
<tr>
<td>JHU, hybrid, FFNN, 2015</td>
<td>4-gram</td>
<td>CDp</td>
<td>4.90</td>
<td>12.98</td>
<td>5.51</td>
</tr>
<tr>
<td>JHU, LF MMI, LSTM, 2016</td>
<td>4-gram</td>
<td>CDp</td>
<td></td>
<td></td>
<td>4.28</td>
</tr>
<tr>
<td>Baidu DeepSpeech2, CTC, 2015</td>
<td>4-gram</td>
<td>chars</td>
<td></td>
<td></td>
<td>5.33</td>
</tr>
<tr>
<td>Facebook, ASG (CTC), 2017</td>
<td>4-gram</td>
<td>chars</td>
<td></td>
<td></td>
<td>4.80</td>
</tr>
<tr>
<td>Salesforce, CTC, PL, 2017</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.10</td>
</tr>
<tr>
<td>attention (this work)</td>
<td>none</td>
<td>BPE</td>
<td>4.87</td>
<td>14.37</td>
<td>4.87</td>
</tr>
<tr>
<td></td>
<td>4-gram</td>
<td>BPE</td>
<td>4.79</td>
<td>14.31</td>
<td>4.82</td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td></td>
<td><strong>3.54</strong></td>
<td><strong>11.52</strong></td>
<td><strong>3.82</strong></td>
</tr>
</tbody>
</table>
Beam Search Error Analysis

- on LibriSpeech, without language model
- reference-related search errors:
  percentage of segments with recognition score worse than reference score:

<table>
<thead>
<tr>
<th>beam size</th>
<th>search errors [%] (WER [%])</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev clean</td>
<td>other</td>
<td>test clean</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.52 (4.87)</td>
<td>1.68 (14.53)</td>
<td>1.07 (4.87)</td>
<td>1.70 (15.49)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.96 (4.88)</td>
<td>0.98 (14.40)</td>
<td>0.76 (4.87)</td>
<td>1.02 (15.39)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.81 (4.87)</td>
<td>0.59 (14.37)</td>
<td>0.61 (4.86)</td>
<td>0.71 (15.39)</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.70 (4.87)</td>
<td>0.52 (14.36)</td>
<td>0.50 (4.86)</td>
<td>0.58 (15.37)</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.26 (4.87)</td>
<td>0.14 (14.34)</td>
<td>0.19 (4.86)</td>
<td>0.20 (15.34)</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- encoder-decoder-attention model for large-vocabulary speech recognition
- target labels: BPE subword units
- pretraining for encoder with time reduction scheduling
- joint beam search with a separately trained LSTM LM
- current experimental results:
  - 300h-Switchboard: competitive results compared to other end-to-end models
    WERs are still higher than the conventional hybrid systems
  - 1000h-LibriSpeech: near to state-of-the-art

- open issues:
  - robustness of training process?
  - how to accommodate lower amounts of training data?
  - need for further model structuring
    → as encoder is similar to networks used in hybrid approach: more robust attention model?
Thank you for your attention

Questions?