1. Introduction

- Language model aims to compute probability of sentence $w_t$
  \[ P(w_t) = P(w_t, w_t+1, w_t+2, \ldots) \]

- Unidirectional language model (uni-LM)
  - The Cat
  - Sat
  - On The Mat

- Estimate $P$(Sat|The Cat)

- Only history information used

- Bidirectional language model (bi-LM)
  - The Cat
  - Sat
  - On The Mat

- Estimate $P$(Sat|The Cat, On The Mat)

- Future word context also used

- Recently, bi-RNNLM outperform uni-RNNLM. However, bi-RNNLM difficult to implement and slow to train

- Difficult for lattice rescoring, n-best rescoring was used

- In this work, su-RNNLM proposed to address these two issues

2. Unidirectional and bidirectional RNNLM

- Unidirectional RNNLM
  - Sigmoid, GRU and LSTM can be used as recurrent units
  - $p(w_t) = \prod_{i=1}^{t} P(w_t|w_{t-i}) \approx \prod_{i=1}^{t} P(w_t|h_{t-i})$

- Bidirectional RNNLM
  - $h_{t-1}$: model past history $w_{t-1}$
  - $h_{t+1}$: model future context $w_{t+1}$
  - $Z = \sum_{w_i \in \Omega} \prod_{t=1}^{L} P(w_t|h_{t-1}, h_{t+1})$
  - Infeasible to compute
  - $P(w_t) \approx \frac{1}{Z} \prod_{t=1}^{L} P(w_t|h_{t-1}, h_{t+1})$

- Unidirectional RNNLM is correct only if
  - a) infinite data, perfect training
  - b) correct history representation

- Bidirectional RNNLM
  - a) product of expert framework
  - b) “optimal” reverse RNNLM

- But, bidirectional RNNLM awkward
  - a) train
  - b) lattice rescoring $\rightarrow$ n-best rescoring used instead

3. Interpolation of LMs

1) Uni-LMs interpolation - linear interpolation
\[
 P_{\text{uni}}(w_k|w_{k-1}) = \lambda P_{\text{ng4}}(w_k|w_{k-1}) + (1 - \lambda) P_{\text{uni}}(w_k|w_{k-1})
\]

2) Bi/Uni-LMs interpolation - log-linear interpolation
\[
 P(w_t^{(i)}) \propto \sum_{\lambda} P_{\text{uni}}(w_t^{(i)}) \cdot P_{\text{bi}}(w_t|w_{t-i}, w_{t+i+1})^{1-\lambda}
\]

4. RNNLM with Succeeding Words (su-RNNLM)

- Recurrent net used for complete history information

- Feedforward net to model a fixed and finite number of succeeding words

5. Lattice Rescoring of su-RNNLMs

- Lattice generated by 2-gram LM

- Lattice rescoring by uni-RNNLM with 3-gram approx.

- Lattice rescoring by su-RNNLM with 3-gram approx. and 1 succeeding word

6. Experimental Results

- Setup
  - AMI IHHM corpus
  - Kaldi recipe for acoustic model construction
  - 14M words for all LM training (4-gram LM, RNNLMs)

- WERs of uni-, bi, and su-RNNLMs with 100-best rescoring.

<table>
<thead>
<tr>
<th>LM</th>
<th>#succ words</th>
<th>train speed</th>
<th>(pseudo) PPL</th>
<th>dev</th>
<th>eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ng4</td>
<td>-</td>
<td>4.5K</td>
<td>80.4</td>
<td>23.8</td>
<td>24.2</td>
</tr>
<tr>
<td>+uni-rnn</td>
<td>-</td>
<td>4.5K</td>
<td>66.8</td>
<td>21.7</td>
<td>22.1</td>
</tr>
<tr>
<td>+su-rnn</td>
<td>-</td>
<td>4.5K</td>
<td>25.5</td>
<td>21.5</td>
<td>21.8</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.9K</td>
<td>21.5</td>
<td>21.3</td>
<td>21.6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.8K</td>
<td>21.3</td>
<td>21.3</td>
<td>21.6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>3.8K</td>
<td>21.3</td>
<td>21.4</td>
<td>21.6</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0.8K</td>
<td>22.1</td>
<td>21.2</td>
<td>21.4</td>
</tr>
</tbody>
</table>

- Training of su-RNNLM is much faster than bi-RNNLM ($\propto$)

- su-RNNLM outperform uni-RNNLM (0.4%-0.5%)

- su-RNNLM slightly worse than bi-RNNLM (0.1%-0.2%)

- WERs of uni- and su-RNNLMs with lattice rescoring

<table>
<thead>
<tr>
<th>LM</th>
<th>#succ words</th>
<th>dev</th>
<th>eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>ng4</td>
<td>-</td>
<td>23.8</td>
<td>23.5</td>
</tr>
<tr>
<td>+uni-rnn</td>
<td>-</td>
<td>21.7</td>
<td>21.5</td>
</tr>
<tr>
<td>+su-rnn</td>
<td>1</td>
<td>21.0</td>
<td>21.3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>21.3</td>
<td>21.4</td>
</tr>
</tbody>
</table>

- Lattice rescoring can be applied on su-RNNLMs

- Consistent improvement obtained from confusion network decoding

- su-RNNLM with 3 succeeding words gave 0.5%-0.6% WER reduction

7. Conclusions

- Future information is useful for language modeling

- Proposed su-RNNLM is easy to implement and fast to train

- Su-RNNLMs suitable for lattice rescoring, consistent WER gain obtained

- Accepted ASRU2017 paper can be found: https://arxiv.org/abs/1708.05592