**Overview**

**Linearization**: given a bag of words, order them into a grammatical sentence.
- Traditional approach uses statistical models
- Recent approaches use LSTMs [1]
  - With or without syntactic linearization (building syntax trees) [2]
- Syntax-free linearizer avoids parsing error and is more lightweight

**Project Goal**: Improve syntax-free neural linearizer using encoders and attention.

**Dataset and Approach**

1) **Dataset = three NLTK corpora**
   - Gutenberg, Brown, Reuters
   - multiple genres & time periods
   - omit sentences with > 20 tokens
   - 96,805 sentences
   - dataset sizes:
     - 1000/10,000/96,805
2) **Input Generation**
   - Split into tokens
     - words + punctuation
     - Randomize order
3) **Run through model**
   - embedding lookup
     - optional encoder
       - with or without attention
       - greedy or beam search
       - with or without random <unk> replacement

**Results and Analysis**

**Experiments:**
- baseline LSTM
- n-layer bidirectional LSTM encoder
- n-layer CNN encoder
- greedy vs. beam search
- w/ vs. w/o <unk> replacement
- w/ vs. w/o attention
- w/ vs. w/o highway layer

**Optimal # of Layers:**
- LSTM: 2
- CNN: 3

**Follow-up Experiments:**
(5 trials on 970 samples)
- CNN Highway: 6.57
- CNN No Highway: 7.51

**CNN-3 yields highest BLEU scores**
- Attention decreases BLEU score on full dataset
- Challenges for the model:
  - rare vocabulary
  - very long sentences

**Summary**

- CNN-3 yields highest BLEU scores
- Attention leads to poorer performance
- LSTM encoder performs similarly to baseline

**Figure 6. Comparison of different models on datasets of varying sizes. CNN-3 without attention performs best.**

<table>
<thead>
<tr>
<th>Data Size</th>
<th>Baseline LSTM</th>
<th>LSTM-2 Encoder</th>
<th>CNN-3 Encoder</th>
<th>CNN-3 Encoder + Bag Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>8.46</td>
<td>9.59</td>
<td>8.89</td>
<td>9.59</td>
</tr>
<tr>
<td>Med</td>
<td>12.42</td>
<td>10.95</td>
<td>15.38</td>
<td>5.65</td>
</tr>
<tr>
<td>Full</td>
<td>20.4</td>
<td>20.19</td>
<td>25.06</td>
<td>4.29</td>
</tr>
</tbody>
</table>

**Future Work**

- Char-LSTM for handling <unk>s
- Transformer model
- Pointer-generator networks

**References**