Parallel Iterative Edit Models for Local Sequence Transduction

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, Vihari Piratla

Correspondence: awasthi@cse.iitb.ac.in @ Awasthi_A

Grammatical Error Correction made 5 to 15 times faster by sequence labeling with \( \pi \)

Standard Approach
Translate incorrect sequence to correct sequence using auto-regressive encoder decoder models

Why explicitly generate the target sequence from scratch?
All we need is a few local edits to the input!

Our Approach
Labeling incorrect sequence with edits

1. Labeling with edits instead of translation
2. Non-autoregressive, parallel predictions
3. Iterative refinement for capturing missed dependencies
4. Rewiring BERT for sequence editing

From translation to sequence labeling with edits

Original Problem: Translation
\[ \mathbf{X} \]: He caught by policeman
\[ \mathbf{Y} \]: He was caught by a policeman

Modification: Sequence Editing
\[ \mathbf{X} \]: He caught by policeman
\[ \mathbf{e} \]: COPY INS\(_{\text{was}}\) REP\(_{\text{caUGHT}}\) COPY

Simplification: Sequence Labeling
Trick: Merge COPY INS\(_{\text{}}\) to form Append\(_{\text{}}\)
\[ \mathbf{X} \]: He caught by policeman
\[ \mathbf{e} \]: Append\(_{\text{was}}\) REP\(_{\text{}}\)

Local Sequence Transduction Problems
1. Grammatical Error Correction
2. Spell Correction
3. OCR – denoising

Key Property: Source and Target Sequence are generally not too different

How fast is sequence labeling w.r.t. translation?

How accurate is sequence labeling w.r.t. translation?

<table>
<thead>
<tr>
<th>Work</th>
<th>Architecture</th>
<th>( F_{0.5} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lichtenberg et al. NAACL 2019</td>
<td>Seq2Seq, Transformers</td>
<td>60.4</td>
</tr>
<tr>
<td>Zhao et al. NAACL 2019</td>
<td>Seq2Seq, Transformers</td>
<td>61.2</td>
</tr>
<tr>
<td>PIE (This work) EACL 2019</td>
<td>Sequence Labeling, Transformers</td>
<td>61.2</td>
</tr>
<tr>
<td>Kyono et al. EMNLP 2019</td>
<td>Seq2Seq, Transformers + Better Pseudo data</td>
<td>65.0</td>
</tr>
</tbody>
</table>

* PIE is expected to provide similar accuracy gains when pre-trained with pseudo data of Kyono et al.

Iterative Refinement of parallelly edited sentences
Input: However, there are two sides of stories always.
Iter1: However, there are always two sides to stories.
Iter2: However, there are always two sides to the story.

Acknowledgements: Google’s TFFRC program for TPUs, Google PhD Fellowship in Machine Learning, Microsoft Student Travel Grant

Abstract
We present a Parallel Iterative Edit (PIE) model for the problem of local sequence transduction arising in tasks like Grammatical error correction (GEC). Recent approaches are based on the popular encoder-decoder (ED) model for seq2seq learning. The ED model autoregressively captures full dependency among output tokens but suffers from sequential decoding, giving up the advantage of modelling full dependency in the output, yet it achieves accuracy competitive with the ED model for four reasons: 1. Labeling sequences with edits instead of generating sequences, 2. Iterative refinement to capture missed dependencies, and 3. Rewiring a pre-trained language model like BERT for edit predictions. Experiments on tasks spanning GEC, OCR denoising and spell correction demonstrate that the PIE model is an accurate and significantly faster alternative.