We adapt our SIGMORPHON 2018 system to Grapheme-to-Phoneme (G2P), a problem with largely disjoint input and output vocabularies.

Highlights

- Neural transducer with copy edit
- Trained with imitation learning [1, IL] and ED(target, prediction) + ED(input, prediction) as loss.

Now:

- Neural transducer with substitution edits
- Trained with IL and ED(target, prediction) + Stochastic ED(input, prediction) as loss. Stochastic ED is a WFST with parameters \( \phi \), which we learn from data.

Model

Just like a WFST, the model monotonically transduces input string \( x \) into output string \( y \) by a sequence of edits \( a \).

- Edits: ins[y] and sub[y] for \( y \in \Sigma_{Y} \), del, copy
- Features: full input context, full edit history

\[
P_s = \text{LSTM}(c_{t-1}, E(E_{t}(c_{t-1}), h_{t}))
\]

\[
P(a_t | a_{<t}, x, \theta) = \text{softmax}(w \cdot s_t)
\]

Example: Model Execution

<table>
<thead>
<tr>
<th>G</th>
<th>a</th>
<th>e</th>
<th>b</th>
<th>x</th>
<th>e</th>
<th>t</th>
<th>s</th>
<th>e</th>
<th>n</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edits</td>
<td>inp</td>
<td>capu</td>
<td>sub</td>
<td>delete</td>
<td>copy</td>
<td>sub</td>
<td>delete</td>
<td>capitalize</td>
<td>sub</td>
<td>delete</td>
</tr>
<tr>
<td>P</td>
<td>a</td>
<td>p</td>
<td>s</td>
<td>s</td>
<td>a</td>
<td>e</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* = whitespace

Stochastic Edit Distance

Our 2018 IL objective ranks edit action sequences by action cost: ED(input, prediction). This addresses spurious ambiguity as multiple edit action sequences achieve the same ED(target, prediction).

For G2P, we use Stochastic Edit Distance (SED), a probabilistic generalization of ED. We learn edit weights \( \phi \) from data with EM.

IL training: Maximize likelihood of edits that are optimal under the loss in any configuration (input, attention pointer, prediction so far, and target).

We find optimal edits on the training set using the SED expert policy.

1. Find permissible edits: They do not increase ED(target, future prediction)
2. Score permissible edits using SED:
   (a) Execute edit
   (b) Edit cost-to-go = edit cost under SED
   (c) Edit cost-to-go + \( \epsilon \) cost of Viterbi path in SED for (input suffix, target suffix).
3. Optimal edits attain lowest cost-to-go

Example: SED policy

- input \( x \) is abject
- target \( y = a_{e}b_{e}c_{e}d_{e}e_{e}f_{e} \)
- attention \( x_{3} = e \), and
- imperfect prediction so far \( \hat{y}_{1,4} = a_{e}b_{e}c_{e}e_{e}f_{e} \)

1. Permissible edits: sub[e], ins[e], del, sub[e], ins[e]
2. Score permissible edits:
   (a) Execute edit: e.g. sub[e] writes \( \epsilon \) and moves the attention to \( x_3 = c \)
   (b) Compute cost-to-go: 15.3 for sub[e], 17.7 for sub[e], 21.1 for ins[e], 17.3 for del, and 17.3 for ins[e]

Qualitative Error Analysis

Table 1: Overview of the test results. \( \Delta \) gives relative error difference compared to our submission CLUZH. \#C = number of NFC models in the ensemble. \#D = number of NFKD models in the ensemble. CLUZH WER AVG = average WER, standard deviation, and relative error difference of the average computed over individual models. \( \Delta \) = lower-bound on WER: correct if predicted by any individual model. LSTM = official seq2seq LSTM baseline. TF = official seq2seq Transformer baseline. BEST = overall best results of other systems for each language.

Setup

Parameters

- input character & action embeddings of size 300 and one-layer LSTMs with hidden-state size 200.
- Training: maximum of 60 epochs with a patience of 12, mini-batches of size 5. Training takes 4 minutes per epoch on CPU (DyNet).

References


Acknowledgements: This work was supported by Swiss National Science Foundation Grant No. CRSII17_17319.