A WFST-based Log-linear Framework for Speaking-style Transformation
Graham Neubig, Shinsuke Mori, Tatsuya Kawahara
Graduate School of Informatics, Kyoto University, Japan

Overview
- **Objective:** Transform spoken-style language (V) into written style language (W) for the creation of transcripts
- **Approach:** Statistical machine translation to "translate" from verbatim text to written text
- **Innovations:**
  - Log-linear modeling for improved accuracy
  - Introduction of features to handle common phenomena in speaking-style transformation
  - WFST-based implementation for integration with WFST-based speech recognizers
- **Evaluation** on transformation of Japanese verbatim transcripts showed improvement over traditional methods

Necessary Transformations

<table>
<thead>
<tr>
<th>Various</th>
<th>AUHT</th>
<th>Things by order</th>
<th>by</th>
<th>make</th>
<th>if</th>
<th>it is</th>
</tr>
</thead>
<tbody>
<tr>
<td>いろんな</td>
<td>こと</td>
<td>で</td>
<td>注文</td>
<td>つける</td>
<td>と</td>
<td>です</td>
</tr>
<tr>
<td>いろいろ</td>
<td>こと</td>
<td>で</td>
<td>注文</td>
<td>つける</td>
<td>と</td>
<td>です</td>
</tr>
</tbody>
</table>

- **Filler Deletions:** Words that are consistently used as fillers: "e-to" "ano-"
- **Other Deletions:** Words that are fillers or not depending on context, repeats, repairs, etc.
- **Substitutions:** Colloquial expressions, etc.
- **Insertions:** Dropped words, particularly particles in Japanese: "o" "wa" "ga"

WER before transformation: 16.40%

Speaking Style Transformation

X

V

W

ASR

SST

that is like uh not a problem

that is not a problem.

Speaking Style Transformation Diagram

- **Noisy Channel Modeling:**
  \[ \hat{W} = \arg \max_w P(W|V) \]
  \[ = \arg \max_w P(V'|W)P(W) \]

- **Log Linear Modeling:**
  \[ \hat{W} = \arg \max_w \log \left( P(V'|W) \right) + \log \left( P(W) \right) \]
  \[ = \arg \max_w \lambda_1 \log \left( P(V'|W) \right) + \lambda_2 \log \left( P(W) \right) + \ldots \]

SST-Specific Features

Extra features can be added to the log-linear model:

\[ \hat{W} = \arg \max_w \lambda_1 \log \left( P(V'|W) \right) + \lambda_2 \log \left( P(W) \right) + \ldots \]

- **Filler Dictionary:** \( f(V,W) \) is equal to the number of fillers (from a 23-word list) present in W.
- **Transformation Groups:** \( f(V,W) \) is equal to the number of groups of words transformed.
  - is like um uh maybe um not a problem

- **Transformation Types:**
  - Insertions, deletions, substitutions are given separate penalties, allowing adjustment of the precision/recall of each type.
- **Decomposed Translation Model:** Use separate log-linear weights for each frequency used when calculating the translation model.

\[
\log P(W|V) = \log \prod_{i=1}^{k} P(v_i,w_i)/P(w_i) \\
= \lambda_1 \log \prod_{i=1}^{k} P(v_i,w_i) - \lambda_2 \log \prod_{i=1}^{k} P(w_i)
\]

Evaluation

Committee meetings of the Japanese National Diet
Verbatim transcripts as input, official transcripts as output
- 3.62M sentences for LM training
- 56.2k aligned sentences for TM training (974 held-out)
- 7181 testing sentences from meetings after the training data

WER before transformation: 16.40%

Context Necessary

<table>
<thead>
<tr>
<th>Fillers</th>
<th>47.3%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-fillers</td>
<td>35.7%</td>
</tr>
<tr>
<td>Sub</td>
<td>8.8%</td>
</tr>
<tr>
<td>Ins</td>
<td>8.2%</td>
</tr>
</tbody>
</table>