Learning NLG Without Alignments

Current NLG systems:
- require alignment of MR elements to words/phrases
- use a separate alignment step

Our generator:
- learns alignment jointly, using just MR + Sentence pairs
- no need for manual annotation/preprocessing
- alignments are latent; MR elements do not need to correspond to words/phrases 1:1

Why?
- no need for manual annotation/preprocessing + cheaper/faster, no error propagation
- alignments are latent: MR elements do not need to correspond to words/phrases 1:1

Scorer

a function:
- sentence plan tree + MR \rightarrow real-valued score
  - describes the fitness of the tree for the MR
- perceptron scorer
  - score = weights \cdot features (from tree and MR)
  - features = elements of tree and MR
  - presence of nodes, slots, values + combination
  - tree size, shape, parent-child
- training loop:
  - given MR, generate the best tree with current weights
  - update weights if generated tree ranks better than gold tree

Overall Schema of Our Generator

Sentence planner
- \textit{A*} search:
  - finding the path from empty tree to full sentence plan tree
  - expand the most promising candidate sentence plan in each step
  - stop when candidates don’t improve for a while

Two sub-components:
- candidate generator – do the expansion
- scorer/ranker – decide what’s promising

Candidate Generator

- create expansions of the sentence plan tree by adding 1 node (at every possible place)
- “possible places” limited by a few simple context rules

Surface realizer

- agreement
- word ordering
- compound verb forms
- grammatical words
- punctuation
- word inflection
- phonetic changes

Plain text sentence

X is an Italian restaurant by the river.

Scorer improvements

- features are global \rightarrow bigger trees tend to score higher
- but we score incomplete trees
- bigger is not always right
- we must promote the “promising” ones
- two improvements to address this:
  1) Differing subtree updates
  2) Future promise estimate

Experiments

- data: restaurant recommendation from the BAGEL generator (Mairesse et al., 2010)
- restaurant location, food type, etc.
- 404 sentences for 202 input dialogue acts, 2 paraphrases each
- manual alignment provided, but we don’t use it
- using 10-fold cross-validation
- measuring BLEU/NIST

Results

- perceptron scorer
  - BLEU 54.24, NIST 4.643
  - + differing subtree updates BLEU 58.70, NIST 4.876
  - + future promise BLEU 59.89, NIST 5.231
  - scorer improvements statistically significant
  - lower scores than Mairesse et al.’s \sim 67% BLEU
- but our problem is harder:
  - we learn alignments jointly
  - our generator has to decide when to stop (whether all required information is included)

Example outputs

<table>
<thead>
<tr>
<th>Setup</th>
<th>Input DA</th>
<th>Reference</th>
<th>Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>perceptron scorer</td>
<td>inform(name=X-name, type=placeoat, pricerange=moderate, eattype=restaurant)</td>
<td>X is a restaurant that offers moderate price range.</td>
<td>X is a restaurant that offers moderate price range.</td>
</tr>
<tr>
<td>+ differing subtree updates</td>
<td>inform(name=X-name, type=placeoat, pricerange=moderate, eattype=restaurant)</td>
<td>X is a moderately priced restaurant in X.</td>
<td>X is a restaurant in the moderate price range.</td>
</tr>
<tr>
<td>+ future promise</td>
<td>inform(name=X-name, type=placeoat, area=K-area, pricerange=moderate, eattype=restaurant)</td>
<td>X is a moderately priced restaurant in X.</td>
<td>X is a restaurant in the X area.</td>
</tr>
</tbody>
</table>

Conclusion

- Learning sentence planning from unaligned data is feasible
- Promising results, but lower than previous with manual alignment (Mairesse et al., 2010)
- outputs mostly fluent and relevant (with some problems)
- Our generator is available at: http://github.com/UFAL-DSG/tgen