Automatic Quality Estimation for Natural Language Generation: Ranting (Jointly Rating and Ranking)

Ondřej Dušek, Karin Sevegnani, Ioannis Konstas & Verena Rieser
Charles University, Prague
Heriot-Watt University, Edinburgh
INLG, Tokyo, 31 Oct 2019
Our Task(s)

• **Quality estimation**: checking NLG output quality
  • just given input MR & NLG system output
  • no human reference texts for the NLG output
  • supervised training from a few human-annotated instances
  • well-established for MT, not so much in data-to-text NLG

• **Rating**: Given NLG output, check if it’s good or not (scale 1-6)

• **Ranking**: Given more NLG outputs, which one is the best?

**MR:** inform_only_match(name='hotel drisco', area='pacific heights')
**NLG output:** the only match i have for you is the hotel drisco in the pacific heights area.

**Rating:**
4 (on a 1-6 scale)

**Rank:**
better

**MR:** inform(name='The Cricketers', eatType='coffee shop', rating=high, familyFriendly=yes, near='Café Sicilia')
**NLG 1:** The Cricketers is a children friendly coffee shop near Café Sicilia with a high customer rating.
**NLG 2:** The Cricketers can be found near the Café Sicilia. Customers give this coffee shop a high rating. It's family friendly.

**Rank:**
better

worse
Why Quality Estimation?

• BLEU et al. don’t work very well – can we be better?
  • evaluating via correlation with humans

• We can do without human references – wider usage:
  • Evaluation, tuning (same as BLEU)
  • Tuning (same as BLEU)
  • Inference – improving running NLG systems

• Inference time use:
  • for rating: don’t show outputs rated below a threshold
    • use a backoff or humans
  • ranking: select best system output from an n-best list
Old Model (Dušek, Novikova & Rieser, 2017)

• Ratings only
• Dual-encoder
  • MR encoder
  • NLG output encoder
  • fully connected + linear
  • trained by squared error
• Final score is rounded
Our Model

- Ranking extension:
  - 2\textsuperscript{nd} copy NLG output encoder + fully connected + linear
    - shared weights
  - trained by hinge rank loss
    - on difference from 2 ratings
- Can learn ranking & rating jointly
  - training instances mixed & losses masked
Synthetic Data (Dušek, Novikova & Rieser, 2017)

- Adding more training instances
  - introducing artificial errors
  - randomly:
    - removing words
    - replacing words by random ones
    - duplicating words
    - inserting random words

- For rating data:
  - lower the rating by 1 for each error (with 6 → 4)

- This can be applied to NLG systems’ training data, too
  - assume 6 (maximum) as original instances’ rating

– articles and punctuation are dispreferred
Synthetic Ranking Pairs

• Different #’s of errors introduced to the same NLG output
• Fewer errors should rank better
• Ranking pairs are useful when the system is trained to rate, too!

restaurant

X-name serves Chinese food .

1 error

Rank: better

food

X-name serves Chinese food .

2 errors

worse

cheaply
Results: Rating

• Small 1-6 Likert-scale data (2,460 instances)
  • 3 systems, 3 datasets (hotels & restaurants)
  • 5-fold cross-validation

• Much better correlations than BLEU et al.
  • despite not needing references
  • synthetic data help a lot
    • statistically significant
  • correlation of 0.37 still not ideal
    • noise in human data?

• absolute differences (MAE/RMSE) not so great

<table>
<thead>
<tr>
<th>System</th>
<th>Pearson</th>
<th>Spearman</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>-</td>
<td>-</td>
<td>1.013</td>
<td>1.233</td>
</tr>
<tr>
<td>BLEU (needs human references)</td>
<td>0.074</td>
<td>0.061</td>
<td>2.264</td>
<td>2.731</td>
</tr>
<tr>
<td>Our previous (Dušek et al., 2017)</td>
<td>0.330</td>
<td>0.287</td>
<td>0.909</td>
<td>1.208</td>
</tr>
<tr>
<td>Our base</td>
<td>0.253</td>
<td>0.252</td>
<td>0.917</td>
<td>1.221</td>
</tr>
<tr>
<td>+ synthetic rating instances</td>
<td>0.332</td>
<td>0.308</td>
<td>0.924</td>
<td>1.241</td>
</tr>
<tr>
<td>+ synthetic ranking instances</td>
<td>0.347</td>
<td>0.320</td>
<td>0.936</td>
<td>1.261</td>
</tr>
<tr>
<td>+ synthetic from systems’ training data</td>
<td>0.369</td>
<td>0.295</td>
<td>0.925</td>
<td>1.250</td>
</tr>
</tbody>
</table>

(Novikova et al., EMNLP 2017)
https://aclweb.org/anthology/D17-1238
Results: Ranking

• Using E2E human ranking data (quality) – 15,001 instances
  • 21 systems, 1 domain
  • 5-way ranking converted to pairwise, leaving out ties
  • 8:1:1 train-dev-test split, no MR overlap

• Our system is much better than random in pairwise ranking accuracy

• Synthetic ranking instances help
  • +4% absolute, statistically significant

• Training on both datasets doesn’t help
  • different text style, different systems

<table>
<thead>
<tr>
<th>System</th>
<th>P@1/Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.500</td>
</tr>
<tr>
<td>Our base</td>
<td>0.708</td>
</tr>
<tr>
<td>+ synthetic ranking instances</td>
<td>0.732</td>
</tr>
<tr>
<td>+ synthetic from systems’ training data</td>
<td>0.740</td>
</tr>
</tbody>
</table>

(Dušek et al., CS&L 59)
https://arxiv.org/abs/1901.07931
Conclusions

• Trained quality estimation can do much better than BLEU & co.
  • Pearson correlation with humans 0.37 vs. ~0.06-0.10
  • synthetic ranking instances help
• The results so far aren’t ideal (we want more than 0.37/74%)
• Domain/system generalization is still a problem
• Future work:
  • improving model
  • using pretrained LMs
  • obtaining “cleaner” user scores
  • more realistic synthetic errors
  • influence of error type on user ratings
Thanks


• Contact me:
  odusek@ufal.mff.cuni.cz
  @tuetschek

  previous model: arXiv: 1708.01759
  datasets used: ACL D17-1238, arXiv:1901.07931