SG01 at the NTCIR-13 STC-2 task

Haizhou Zhao, Yi Du, Hangyu Li, Qiao Qian, Hao Zhou, Minlie Huang, Jingfang Xu

Sogou Inc. | Beijing, China
Tsinghua University | Beijing, China
Introduction

- Team Name: **SG01**
  - Joint Team from **Sogou Inc.** and **Tsinghua University**

- Subtask: **Chinese subtask**
  - Retrieval-based method: 3 submissions
  - Generation-based method: 5 submissions
  - **Top performance** in both methods

- Next ...
  - Retrieval-based Method
  - Generation-based Method
  - Conclusions
  - Q & A
Overview of Retrieval-based Method

Retrieve Stage

Ranking Stage I

Ranking Stage II

Learn to rank

query

repo

500 pairs

50 pairs

10 pairs

features
Retrieve Stage | Retrieval-based Method

- Data-Preprocessing
  - Remove *frequent, advertising* and *short* post-comment pairs
- Put the repository into a light-weighted search engine
  - Treat post-comment pairs as webpages
- Retrieve **500 pairs** for a given query (or “new post”)
- Keep the calculated **features for searching** for later usage
  - BM25
  - MRF for term dependency [D. Metzler 2005]
  - Proximity [T. Tao 2007]
  - ...
Ranking Stage I | Retrieval-based Method

- Employ **features more intuitive in STC task**
  - **cosine similarity** of TF-IDF vector between ...
  - negative **Word Mover Distance** [M. J. Kusner 2015] between ...
    - query $\leftrightarrow$ post
    - query $\leftrightarrow$ comment
    - query $\leftrightarrow$ post + comment
  - **Translation based language model** [Z. Ji 2014] $Score_{trans}$

- **Ranking**
  - Treat each feature as a ranker
  - Simply add the sequence numbers to get a final rank
  - Keep **top 50 pairs**
Employ more neural network features capturing richer structure in STC

- $Score_{embd}$
- $Score_{BiLSTM+CNN}$ [R. Yan 2016]

\[
L = \max(0, 1 - s(x, y^+) + s(x, y^-))
\]

Trained with ranking based objective, using given repository plus extra 12 million crawled post-comment pairs, noted as $Repo_{extn}$

- $Score_{S2S-p2c}$ ← Defined later in Generation-based Method
- $Score_{S2S-c2p}$
- Use **all features aforementioned**
- Training data: given 11 thous. plus 30 thous. labeled pairs
- **LambdaMART**
- **Top 10** to be the final result
- $Score_{trans}$ and $Score_{BiLSTM+CNN}$ are a little more important
### Experiments

<table>
<thead>
<tr>
<th>Submission</th>
<th>Learning to rank respect to which measure on training data</th>
<th>nG@1</th>
<th>P+</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG01-C-R1</td>
<td>nG@1</td>
<td>0.5355</td>
<td>0.6084</td>
<td>0.6579</td>
</tr>
<tr>
<td>SG01-C-R2</td>
<td>nERR@10</td>
<td>0.5168</td>
<td>0.5944</td>
<td>0.6461</td>
</tr>
<tr>
<td>SG01-C-R3</td>
<td>P+</td>
<td>0.5048</td>
<td><strong>0.6200</strong></td>
<td><strong>0.6663</strong></td>
</tr>
</tbody>
</table>
Overview of Generation-based Method

Generative Models

- S2SAttn + addmem
- VAEAttn + addmem

Segment-beam-search decoding

candidates

Scoring & Ranking

10 pairs
Generative Models | Generation-based Method

- $S2SAttn$
  - seq2seq [I. Sutskever 2014] with attention mechanism
- $S2SAttn$–$addmem$
  - Add dynamic memory to the attention
- $VAEAttn$
  - Use Variational Auto-Encoder
- $VAEAttn$–$addmem$

Training data: $Repo_{extn}$ with data-preprocessing

Decode using $segment$-$beam$-$search$
 Candidates Ranking: scores | Generation-based Method

- Scoring Features
  - **likelihood**
    - $\log(P(Y'|X))$, for post $X$ and generated comment $Y'$
  - We note score from one model as $Score_{s2s-p2c}$
  - For scores from different models (except VAE models) and implementations, we add them up as $Li$

- **posterior**
  - $\log(P(X|Y'))$
  - $Score_{s2s-c2p}$
  - $Po$

- Calculated by our **well trained models**
Candidates Ranking: rank & output

- **Ranking**
  
  $$score = \frac{\lambda \cdot Li + (1-\lambda) \cdot Po}{lp(Y')}$$

- Discount factor $lp(Y') = \frac{(c+|Y'|)^\alpha}{(c+1)^\alpha}$ [Y. Wu 2016]

- **Before Final Output: Process candidates by rules**
  - Abandon candidates with keywords in blacklist
  - De-duplicate **consecutively repeated segments**
  - Truncate **consecutively repeated punctuations**
## Experiments | Generation-based Method

<table>
<thead>
<tr>
<th>Submission</th>
<th>Fusion of candidates from*</th>
<th>Scoring By**</th>
<th>nG@1</th>
<th>P+</th>
<th>nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG01-C-G5</td>
<td>VAEAttn, VAEAttn–addmem</td>
<td>Li</td>
<td>0.3820</td>
<td>0.5068</td>
<td>0.5596</td>
</tr>
<tr>
<td>SG01-C-G4</td>
<td>S2SAAttn, S2SAAttn–addmem</td>
<td>Li</td>
<td>0.4483</td>
<td>0.5545</td>
<td>0.6129</td>
</tr>
<tr>
<td>SG01-C-G3</td>
<td>S2SAAttn, S2SAAttn–addmem</td>
<td>Li &amp; Po</td>
<td>0.5633</td>
<td>0.6567</td>
<td>0.6947</td>
</tr>
<tr>
<td>SG01-C-G2</td>
<td>VAEAttn, VAEAttn–addmem</td>
<td>Li &amp; Po</td>
<td>0.5483</td>
<td>0.6335</td>
<td>0.6783</td>
</tr>
<tr>
<td>SG01-C-G1</td>
<td>All 4 kinds of models</td>
<td>Li &amp; Po</td>
<td><strong>0.5867</strong></td>
<td><strong>0.6670</strong></td>
<td><strong>0.7095</strong></td>
</tr>
</tbody>
</table>

*: could be multiple implementations for one model, using different subset of corpus and hyper-parameters

**: all scores are discounted by lp
The feature $Po$ brings advantage with statistical significance to those without $Po$, by giving higher rank to more informative candidates.

$VAE$ does worse than traditional seq2seq, but it can bring in interesting candidates.

Using fusion of results from models do better than relying on single model, because the ranking will bring preferable candidates to top 10.
Conclusions

- Comparison between methods
  - **Generation-based method** does better, however, it still tends to generate “safe” responses
  - Retrieval-based method tends to get context-dependent or incoherent comments
- Size of training data matters


Thank you!

zhaohaizhou@sogou-inc.com