The splab at the NTCIR-12 STC Task

Ke Wu, Xuan Liu and Kai Yu

Emails: { ke.wu, 1097114522, kai.yu }@sjtu.edu.cn
Motivation

• Similar to finding a needle in a haystack, it is hard to obtain a proper response in an extremely large conversation data.

• Too many irrelevant candidate comments can potentially hinder a system's ability to identify the appropriate response from the large pool of candidates.
Basic Idea

• To collect sufficient candidate comments including the suitable responses in the small candidate pool.

• Our system attempts to facilitate high short text conversation performance by a three-level ranking framework.
  – Through a couple of selection turns, we generate the final plausible candidate set.
  – In the small pool of candidates, we leverage deep learning techniques to find the best response.
Our three-tier ranking framework

- Short text
- Short text analysis
- Retrieval
- Surface feature extraction
- SVM Ranking
- Semantic Ranking
- Best response

Post-comment pairs

Search

Lexical Ranking

Distributed representations of comments
Search Component

- Short text analysis
  - Tailor a short text to use IR technologies
  - Translate a short text into multiple terms with weights

- Retrieval
  - To index all post-comment pairs for a highly-efficient information filtering
Short text analysis

- **Method 1**: MG
  - all potential words with equal weights
  - to improve search recall

- **Method 2**: TFIDF
  - keywords by TF-IDF

- **Method 3**: TextRank
  - Similar to Google’s PageRank algorithm

Detect or recognize the focus with salient information in a short text
Retrieval

• To index all post-comments pairs (offline)

* To use the default similarity function in Lucene (online)

\[
\text{score}(q, s) = \text{co}(q, s) \cdot \text{qn}(q) \cdot \sum_{t \in q} \{ \text{tf}(t \in s) \cdot \text{idf}(t)^2 \cdot w_t \cdot \text{norm}(t, s) \}
\]
**Lexical Ranking Component**

**Goal:** To try to promote all the relevant comments to the top of a ranked list based on the downstream comment candidates by leveraging a small portion of labelled post-comment pairs.

**Lexical Features:**

- Similarity features
- Matching features
  - Longest common string and co-occurring statistics

**Ranking SVM:**

Used to exploit lexical features to ranking the candidates
# Lexical Features

<table>
<thead>
<tr>
<th>Types</th>
<th>Features</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Similarity Features</td>
<td>Q2C</td>
<td>Similarity between the query $q$ and the candidate comment $c$</td>
</tr>
<tr>
<td></td>
<td>Q2P_Ave</td>
<td>Average of the similarities between the query $q$ and the posts with which the candidate comment $c$ is paired</td>
</tr>
<tr>
<td></td>
<td>Q2P_Max</td>
<td>Maximum of the similarities between the query $q$ and the posts with which the candidate comment $c$ is paired</td>
</tr>
<tr>
<td></td>
<td>Q2P_Min</td>
<td>Minimum of the similarities between the query $q$ and the posts with which the candidate comment $c$ is paired</td>
</tr>
<tr>
<td>Matching Features</td>
<td>LCS</td>
<td>Length of the longest common string between the query $q$ and the candidate comment $c$</td>
</tr>
<tr>
<td></td>
<td>LCS_Rate</td>
<td>Ratio of LCS to the length of the candidate comment</td>
</tr>
<tr>
<td></td>
<td>Co_Size</td>
<td>Number of co-occurring words between the query $q$ and the candidate comment $c$</td>
</tr>
<tr>
<td></td>
<td>Co_Rate</td>
<td>Ratio of Co_Size to the number of words in the candidate comment $c$</td>
</tr>
<tr>
<td></td>
<td>Co_IDF_Sum</td>
<td>Sum of IDF values of co-occurring words between the query $q$ and the candidate comment $c$</td>
</tr>
<tr>
<td></td>
<td>Co_IDF_Ave</td>
<td>Average of IDF values of co-occurring words between the query $q$ and the candidate comment $c$</td>
</tr>
<tr>
<td>Others</td>
<td>Post_Num</td>
<td>Number of the posts with which the candidate comment $c$ is paired</td>
</tr>
</tbody>
</table>
Semantic Ranking Component

• Receive and re-rank the aggregated results of three different search strategies from the lexical ranking component by semantics
Our Sentence Embedding

Take max operation at each dimension

Semantic layer
Max-pooling layer
Abstraction layer
Word n-gram layer

<s> 上海 是 中国 的 大城市 之一 。 <s> Short Text
Submitted Results

<table>
<thead>
<tr>
<th>Run name</th>
<th>Mean nDCG@1</th>
<th>Mean P+</th>
<th>Mean nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>splab-C-R1</td>
<td>0.2933</td>
<td>0.4735</td>
<td>0.4449</td>
</tr>
<tr>
<td>splab-C-R2</td>
<td>0.0967</td>
<td>0.2069</td>
<td>0.1831</td>
</tr>
<tr>
<td>splab-C-R3</td>
<td>0.0967</td>
<td>0.1896</td>
<td>0.1650</td>
</tr>
</tbody>
</table>

- splab-C-R1 uses the three-tie framework
- splab-C-R2 uses the first method on RHS
- splab-C-R3 uses the third method on RHS
Conclusions

- we described our system's three-pronged strategy for identifying proper responses that balance high candidate recall and processing time for candidate scoring.

- The evaluation on a test set of 100 test queries provided by the organizers shows that our three-tier ranking system is effective.
THANK YOU
for your
ATTENTION!