**Abstract**

The splab team participated in the Chinese subtask of the NTCIR-12 on Short Text Conversation Task. This task assumes that the existing comments in a post-comment repository can be reused as suitable responses to a new short text. Our task is to return 10 most appropriate comments to such a short text. In our system, we attempt to employ advanced IR methods and the recent deep learning techniques to tackle the problem. We develop a three-tier ranking framework to promote the most suitable comments in top position as much as possible. It consists of three components, i.e., search, lexical ranking and semantic ranking. In the search component, three different query generation methods are employed to boost the system’s recall. In the lexical ranking, we exploit the training data of labelled post-comment pairs to score the comments in the candidate pool. In the semantic ranking, we apply the deep learning techniques to convert the comment string or a short text string to a continuous, low-dimensional feature vector, re-score the nal candidate comments and provide the 10 most reasonable comments to a short text. The evaluation of submitted results empirically shows our framework is effective in terms of mean nDCG@1, mean P+ and mean nERR@10.

**Our System**

**Step 1: Search component**
- To convert a short text to query terms and attempt to retrieve relevant contents as initial candidate comments
  1. Short text analysis (a short text -> query terms)
     * Method 1: all potential words with equal weights
     * Method 2: keywords by TF-IDF
     * Method 3: keywords by TextRank
  2. Retrieval
     * To index all post-comments pairs (offline)
     * To use the default similarity function in Lucene (online)

**Step 2: Lexical ranking component**
- 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.
  1. Surface feature extraction
     * Similarity features
       * Longest common string and co-occurring statistics
  2. SVM ranking
     * To use the features from the surface feature extraction module
     * To apply the ranking SVM by Thorsten Joachims

**Step 3: Semantic ranking component**
- Receive and re-rank the aggregated results of three different search strategies from the lexical ranking component by semantics

**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>

Evaluation results of Chinese short text conversation task for three runs

**Conclusions**

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

**Acknowledgement**

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61573241 and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China.