Microsoft Research Asia at NTCIR-12 STC Task

Zhongxia Chen\textsuperscript{2}, Ruihua Song\textsuperscript{1} and Xing Xie\textsuperscript{1}

\textsuperscript{1} Social Computing Group, Microsoft Research Asia
\textsuperscript{2} University of Science and Technology of China
Motivations

• Similar comment?
  – May be relevant but repetitive

| Test Post   | 看这几天天气挺好的，到上海去玩一圈  
Since the weather is good these days, I'd like to go to Shanghai for a visit |
| Comment1    | 到上海去玩?  
Going to Shanghai? |
## Motivations

• **Similar post?**

  – Not bad

| Test Post | 看这几天天气挺好的，到上海去玩一圈  
Since the weather is good these days, I'd like to go to Shanghai for a visit |
|-----------|-----------------------------------------------------------------|
| Comment1  | 到上海去玩?  
Going to Shanghai? |
| Similar Post | 看这几天天气挺好的，到北京去玩一圈  
Since the weather is good these days, I'd like to go to Beijing for a visit |
| Comment2  | 羡慕  
Envying you |
Motivations

• Similar post?
  
  – But perhaps too specific

| Test Post                                      | 看这几天天气挺好的，到上海去玩一圈  
Since the weather is good these days, I'd like to go to Shanghai for a visit |
|-----------------------------------------------|------------------------------------------------------------------------|
| Comment1                                      | 到上海去玩?  
Going to Shanghai?                     |
| Similar Post                                  | 看这几天天气挺好的，到北京去玩一圈  
Since the weather is good these days, I'd like to go to Beijing for a visit |
| Comment2                                      | 羡慕  
Envying you                                      |
| Comment3                                      | 推荐去颐和园  
I recommend you to go to the Summer Palace |
Our Ideas

• Two observations
  – The comments responded to similar posts are good candidates
  – The more common a comment is, the less risky
System Description

- **Preprocessing**
  - Stanford Chinese Word Segmenter
  - Filter stop words

<table>
<thead>
<tr>
<th>Short Text ID</th>
<th>repos-cmnt-1000578260</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Origin Text</strong></td>
<td>西班牙踢球多黏啊总看传球了’意大利必胜 (Spain always passes and I will support Italy)</td>
</tr>
<tr>
<td><strong>Segmentation Result</strong></td>
<td>西班牙 踢球 多 黏 啊 总 看 传球 了 ’ 意大利 必胜</td>
</tr>
<tr>
<td><strong>Filtering Result</strong></td>
<td>西班牙 踢球 黏 总 传球 意大利 必胜</td>
</tr>
</tbody>
</table>
System Description

Feature Generation

- Post-Post Similarity
  - Generate word vectors
    - Google Word2Vec
    - External Chinese Corpus
  - Generate short text vectors
  - Summation of word vectors
  - Calculate similarity
    - Cosine similarity
      \[ F_{sim}(p, Ci) = \frac{\mathbf{v}_p \cdot \mathbf{v}_{Ci}}{|\mathbf{v}_p| \cdot |\mathbf{v}_{Ci}|} \]
System Description

Feature Generation

Popularity

- Count frequencies of comments
- Top ten comments in repository

<table>
<thead>
<tr>
<th>Comment</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>哈哈(laug)</td>
<td>14830</td>
</tr>
<tr>
<td>哈哈哈哈哈(laug)</td>
<td>8096</td>
</tr>
<tr>
<td>呵呵(no comment)</td>
<td>8075</td>
</tr>
<tr>
<td>嗯(fine)</td>
<td>4443</td>
</tr>
<tr>
<td>哈哈哈哈(laug)</td>
<td>4297</td>
</tr>
<tr>
<td>不错(not bad)</td>
<td>4222</td>
</tr>
<tr>
<td>好(good)</td>
<td>3676</td>
</tr>
<tr>
<td>是的(yes)</td>
<td>3472</td>
</tr>
<tr>
<td>喜欢(I like it)</td>
<td>3216</td>
</tr>
<tr>
<td>赞(great)</td>
<td>3209</td>
</tr>
</tbody>
</table>
System Description

Feature Generation

- Popularity
  - Count frequencies of comments
  - Considering Power Law
  - Generate popularity feature by log function of frequency

\[ F_{pop}(Ci) = \log(Frequency(Ci)) \]
System Description

- **Ranking**
  - Retrieve top k comments with closed corresponding posts using similarity feature, then re-rank by popularity in a small range.
  - Train a linear model
    \[
    \text{Score}(p, Ci) = F_{sim}(p, p') + w \cdot F_{pop}(Ci)
    \]
    where, Ci is a comment of p’. Then rank by score.
Experiments

• Submitted three runs
  – MSRSC-C-R1: ranking by a linear combination of post-post similarity and popularity
  – MSRSC-C-R2: retrieving top 50 comments by post-post similarity and re-ranking by popularity
  – MSRSC-C-R3: ranking by post-post similarity

• Word2vec settings
  – Skip-gram model
  – Window size is 10
  – Vector length is 100
Experiments

• Result for 3 runs

<table>
<thead>
<tr>
<th>Run</th>
<th>Mean nDCG@1</th>
<th>Mean P+</th>
<th>Mean nERR@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRSC-C-R1</td>
<td>0.3367</td>
<td>0.4854</td>
<td>0.4592</td>
</tr>
<tr>
<td>MSRSC-C-R2</td>
<td>0.2733</td>
<td>0.4208</td>
<td>0.3857</td>
</tr>
<tr>
<td>MSRSC-C-R3</td>
<td>0.0933</td>
<td>0.2420</td>
<td>0.2236</td>
</tr>
</tbody>
</table>

– Population is surprisingly helpful
– Linear model is better than retrieval and re-ranking model
  • Range of best fifty similar posts is uncertain
Conclusions

• Post-post similarity is useful
  – We use word embedding in representing short texts

• Popularity of comments is helpful when being combined with post-post similarity
Thank you!

Contact: rsong@microsoft.com