Abstract

Yake! is a novel feature-based system for multi-lingual keyword extraction, which supports texts of different sizes, domain or languages. Unlike other approaches, Yake! does not rely on dictionaries nor thesauri, neither is trained against any domain. Instead, it follows an unsupervised approach which builds upon features extracted from the text, making it thus applicable to documents written in different languages without the need for further knowledge. This can be beneficial for a large number of tasks and a plethora of situations where the access to training corpora is either limited or restricted.

Keyword Extraction Process

- **Raw Text**
  - Sentence Split;
  - Tokenization;
  - Tag words:
    - Parsable word;
    - Upper case;
    - Numbers;
    - Stopwords.

- **Step 1: Preprocessing**
  - Set the term features:
    \[ W_{TF} = \frac{\max(TF(u), TF(w))}{\log(TF(w))} \]
    \[ W_{DPSEN} = \frac{SF(w)}{\#Sentences} \]
    \[ W_{POS} = \frac{TF(w)}{\text{Median}(TF(w) + \text{STD})} \]
    \[ W_{rel} = (\frac{3 \times W_{TF}}{\text{Max}(TF) + 1}) + (\frac{3 \times W_{TF}}{\text{Max}(TF) + PR}) \]
  - Score each term by:
    \[ S(w) = \frac{W_{Rel} \times W_{Position}}{W_{Class} + W_{DFSEN} + W_{Rel}} \]

- **Step 2: Feature Extraction**
  - Extract all n-grams (with n=1,2,3);
  - Remove keywords that begin and/or end with stopword;
  - Score candidates by:
    \[ S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) + (1 + \sum_{w \in kw} S(w))} \]

- **Step 3: Generate and Score Candidates**
  - Rank keywords based on S(kw) ascending order;
  - Deduplicate the keywords using Levenshtein > 0 (where θ=0.9).

- **Step 4: Keyword Deduplication and Ranking**
  - List of Keywords

Results

<table>
<thead>
<tr>
<th>Method</th>
<th>P@10</th>
<th>R@10</th>
<th>F@10</th>
<th>P@10</th>
<th>R@10</th>
<th>F@10</th>
<th>P@10</th>
<th>R@10</th>
<th>F@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>YAKE!</td>
<td>0.153</td>
<td>0.103</td>
<td>0.123</td>
<td>0.217</td>
<td>0.056</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TextRank</td>
<td>0.101*</td>
<td>0.067*</td>
<td>0.081*</td>
<td>0.198*</td>
<td>0.052*</td>
<td>0.082*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF.IDF [4]</td>
<td>0.036*</td>
<td>0.023*</td>
<td>0.028*</td>
<td>0.100*</td>
<td>0.028*</td>
<td>0.043*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SingleRank [5]</td>
<td>0.035*</td>
<td>0.022*</td>
<td>0.027*</td>
<td>0.082*</td>
<td>0.024*</td>
<td>0.037*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAKE [6]</td>
<td>0.007*</td>
<td>0.004*</td>
<td>0.005*</td>
<td>0.013*</td>
<td>0.004*</td>
<td>0.006*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conclusions

- In this work, we propose an unsupervised lightweight approach for keyword extraction of a single document;
- YAKE! achieves better results in comparison to four state-of-the-art unsupervised keyword extraction algorithms;
- Future works
  - Investigate how our method performs in comparison with other datasets and other unsupervised and supervised algorithms;

References