WRT-1017: Keyphrase Extraction using Language Embeddings

Sponsor: US Army CCDC

Dr. K.P. (Suba) Subbalakshmi

12th Annual SERC Sponsor Research Review
HOSTED VIRTUALLY ON: November 18, 2020
www.sercuarc.org
AI Enabled Keyphrase Extraction

- **Keyphrase extraction** is the first step for a lot of downstream NLP tasks needed for competitive intelligence
  - Select relevant articles for SME attention
  - Rank ideas according to importance
  - Extract trends
  - Discover areas of interest
  - Create summaries of documents
Goal of work

- **Goals:** propose new keyphrase extraction algorithms that meets or exceeds state-of-the-art (SOTA) methods
  - Balance diversity of keyphrases with relevance
  - Use title of article to inform keyphrase extraction
  - Use language embedding to represent keyphrases, documents and title
  - Suggest novel make-up of keyphrases
Innovation of our algorithm

• **Title Similarity:**
  — Take advantage of relationship between the article title and keyphrases

• **Frequency:**
  — Keyphrases are likelier to appear more often than other types of words
  — Select these phrases as candidates

• **Candidate keyphrases:**
  — Suggest new structure for keyphrases

• **New Metric (incorporating diversity and title information):**
  — New metric proposed to trade-off relevance of keyphrases with the similarity *between* keyphrases
Proposed Architecture

- **d_doc**: distance between candidate phrase & document
- **d_p**: distance between candidate phrases
- **d_t**: distance between candidate phrase and title

Compute metric:
\[ \beta \times (d_{doc} + d_t) + (1 - \beta)d_p \]

Rank and choose

Universal Sentence Embedder

Candidate phrase selector

Calculate high frequency phrases and output directly
Proposed Method: Extracting candidate phrases

- Extract candidate phrases from the text
  - Keyphrases, (as opposed to keywords) always consist of two or more words and have some fixed part-of-speech types

- Proposed make-up of candidate keyphrases
  - Adj(one or more) + Nouns (One or more)
  - Nouns (one or more) + Nouns (one or more)
  - VBN + Nouns → example: “distributed system”
  - Adj + VBG + Nouns → example: “minimal generating sets”
  - Adj + VBN + Nouns → example: “out-of-print print music”

- *EmbedRank uses Adj (zero or more) + Nouns (One or more)*
First step (contd.) : Extract candidate phrases

• Delete unimportant words for candidate phrase
  — e.g: Such More Other Most Less Few Little Many Better

  — Example of trimmed phrases:
    o ['such information architecture'] → ['information architecture']

• Compute high frequency keyphrases (that occur two or more times) and output them directly as predicted keyphrases
Third step: Language Embedding

- Use sentence embedding to represent:
  - the candidate phrases
  - the title
  - the document in the same high-dimensional vector space

- We use the Universal Sentence Encoder (USE)
  - Proposed by Google and uses either the transformer architecture or the deep averaging network
  - Considered the SOTA for language embedding
  - Produces a 512 dimension vector for each phrase/document/sentence
• EmbedRank (one of the comparison algorithms) -
  — an unsupervised method to automatically extract keyphrases from a document
  — Uses sent2vec as embedding method for sentences

Step 1
Extract candidate phrases

Step 2
Embed candidate phrases and the document

Step 3
Rank and Select

\[ \beta \ast \text{sim}_{to\_doc} - (1-\beta)\ast \text{sim}_{between}. \]
The Metric

- We propose a new metric to trade-off the relevance of the keyphrase against the similarity between the keyphrases
  - Ensures diversity and relevance of keyphrases

- $\beta^* (\text{norm}_\text{sim}_\text{doc} + \text{norm}_\text{sim}_\text{title}) + (1-\beta)^* \text{norm}_\text{sim}_\text{between}$
  - $\text{norm}_\text{sim}_\text{doc}$: normalized cosine similarity between the document and each candidate phrase
  - $\text{norm}_\text{sim}_\text{title}$: normalized cosine similarity between the title and each candidate phrase
  - $\text{norm}_\text{sim}_\text{between}$: normalized cosine similarity between candidate phrases
  - $\beta$: Controls the relative importance between relevance and diversity.
Rank and Select Keyphrases

• The score for each candidate keyphrase is calculated using the proposed metric

• Keyphrases are ranked and the top N keyphrases are selected

• Evaluation Metrics:
  — We evaluate our method against two most recent methods
  — Using two different metrics
Evaluation Method 1:

Precision: \# Matching predicted keyphrases / Total \# predicted keyphrases

Recall: \# Matching predicted Keyphrases / Total \# ground-truth keyphrases

F1 score: \(2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall})\)
## Comparisons

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>Inspec</th>
<th>Nus</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Embedrank</td>
<td>31.51</td>
<td>10.28</td>
</tr>
<tr>
<td></td>
<td>Multipartite</td>
<td>25.78</td>
<td>13.31</td>
</tr>
<tr>
<td></td>
<td><strong>Ours</strong></td>
<td><strong>34.18</strong></td>
<td><strong>16.32</strong></td>
</tr>
<tr>
<td>10</td>
<td>Embedrank</td>
<td>37.94</td>
<td>13.43</td>
</tr>
<tr>
<td></td>
<td>Multipartite</td>
<td>30.01</td>
<td>16.92</td>
</tr>
<tr>
<td></td>
<td><strong>Ours</strong></td>
<td><strong>40.64</strong></td>
<td><strong>17.86</strong></td>
</tr>
<tr>
<td>15</td>
<td>Embedrank</td>
<td>38.40</td>
<td>14.47</td>
</tr>
<tr>
<td></td>
<td>Multipartite</td>
<td>31.44</td>
<td>17.16</td>
</tr>
<tr>
<td></td>
<td><strong>Ours</strong></td>
<td><strong>42.29</strong></td>
<td><strong>18.67</strong></td>
</tr>
</tbody>
</table>

*Multipartite*: The state-of-the-art method for the NUS dataset mentioned in the Embedrank paper.
Evaluation Method 2

- **Precision:** \( \frac{\text{# Matching predicted keyphrases}}{\text{total # predicted keyphrases}} \)

- **Recall:** \( \frac{\text{# Matching predicted keyphrases}}{\text{total # ground-truth keyphrases present (Ignore the absent keyphrases)}} \)

- **F1 score:** \( \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} \)
  - Metric proposed in: “One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases (ACL2020)”
## Evaluation Result: Compare to state-of-the-art method

<table>
<thead>
<tr>
<th></th>
<th>Inspec</th>
<th>Krapivin</th>
<th>NUS</th>
<th>SenEval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>F1--5</td>
<td>F1--10</td>
<td>F1--X</td>
<td>F1--5</td>
</tr>
<tr>
<td>CopyRNN</td>
<td>24.4</td>
<td>28.9</td>
<td>29</td>
<td>30.5</td>
</tr>
<tr>
<td>KG-KE-KR-M</td>
<td>25.7</td>
<td>28.4</td>
<td>31.4</td>
<td>27.2</td>
</tr>
<tr>
<td>CateSeqD</td>
<td>27.6</td>
<td>33.3</td>
<td>33.1</td>
<td>32.5</td>
</tr>
<tr>
<td>Abstractive Neural</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DivGraphPointer</td>
<td>38.6</td>
<td>41.7</td>
<td>46.2</td>
<td>40.2</td>
</tr>
</tbody>
</table>

### Extractive Neural

<table>
<thead>
<tr>
<th></th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-Multi</td>
<td>32.8</td>
<td>31.8</td>
<td>32.3</td>
<td>25.4</td>
<td>36.5</td>
<td>32.6</td>
<td>31.9</td>
<td>31.2</td>
<td></td>
</tr>
<tr>
<td>TG-Net</td>
<td>31.5</td>
<td>38.1</td>
<td>34.9</td>
<td>29.5</td>
<td>40.6</td>
<td>37</td>
<td>31.8</td>
<td>32.2</td>
<td></td>
</tr>
</tbody>
</table>

### Extractive IR

<table>
<thead>
<tr>
<th></th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
<th>F1--5</th>
<th>F1--10</th>
<th>F1--X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tfidf</td>
<td>16</td>
<td>24.4</td>
<td>20.8</td>
<td>6.7</td>
<td>9.3</td>
<td>6.8</td>
<td>11.2</td>
<td>14</td>
<td>12.2</td>
</tr>
<tr>
<td>TextRank</td>
<td>28.6</td>
<td>33.9</td>
<td>33.5</td>
<td>33.5</td>
<td>16</td>
<td>33.5</td>
<td>23</td>
<td>21.6</td>
<td>23.8</td>
</tr>
<tr>
<td>KEA</td>
<td>2.2</td>
<td>2.2</td>
<td>2.2</td>
<td>1.8</td>
<td>1.7</td>
<td>1.7</td>
<td>7.3</td>
<td>7.1</td>
<td>8.1</td>
</tr>
<tr>
<td>Maui</td>
<td>3.5</td>
<td>4.6</td>
<td>3.9</td>
<td>0.5</td>
<td>0.7</td>
<td>0.6</td>
<td>0.4</td>
<td>0.6</td>
<td>0.6</td>
</tr>
<tr>
<td>Ours</td>
<td>39.12</td>
<td>44.51</td>
<td>45.25</td>
<td>25.78</td>
<td>23.45</td>
<td>26.8</td>
<td>29.48</td>
<td>27.43</td>
<td>30.09</td>
</tr>
</tbody>
</table>

Here X denotes the number of ground-truth keyphrases present in the original document.
Conclusion

- **INSPEC**: Our method is better than all models: supervised methods and unsupervised methods.

- **NUS, Krapivin and SemEval**: Our method is better than all unsupervised methods. And it is close to some supervised models and better than some supervised methods.
## Result analysis

<table>
<thead>
<tr>
<th>Datasets</th>
<th>INSPEC</th>
<th>Semeval</th>
<th>Karpivin</th>
<th>NUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of ground-truth keyphrases</td>
<td>3858</td>
<td>671</td>
<td>1485</td>
<td>1263</td>
</tr>
<tr>
<td>number of single word ground-truth keyphrases</td>
<td>531(13.76%)</td>
<td>161(23.99%)</td>
<td>307(20.67%)</td>
<td>409(32.38%)</td>
</tr>
<tr>
<td>number of overlapping ground-truth keyphrases</td>
<td>10</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Ground-truth keyphrases: ground-truth keyphrases present in the original document.

Because our method focuses on keyphrases which have two or more words, we get significant improvements over other methods for INSPEC.