Neural Extractive Text Summarization with Syntactic Compression

Adib Mosharrof
Overview

- Create a textual summary of a single document
- Produce summary that is syntactically and grammatically correct
- Used single document news summarization datasets
  - CNN
  - Daily Mail
  - New York Times
- Can produce different length summaries
Figure 1: Diagram of the proposed model. Extraction and compression are modularized but jointly trained with supervision derived from the reference summary.
Extraction Module: Encoding

- Use a bidirectional LSTM to encode words in each sentence
- Aggregate sentence representation into document representation $v_{doc}$
- Multiple convolutional and max pooling layers applied
Decoding

- Selects sentences to include in summary
- Sequential LSTM decoder used
- Selection is similar a pointer network
  - variation of the sequence-to-sequence model with attention
  - succession of pointers to the elements of the input series

Decoder state \((d_t)\)

\(W_d, W_h, W_m\) are parameters learned in LSTM

\(H_i\) is the sentence representation
Text Compression

Decides whether to remove certain phrases or words in selected sentences

Uses a feedforward network for decision

A contextualized encoder ELMo, is used to compute contextualized word representations

CNN with max pooling used to encode sentence (shown in blue) and the candidate compression (shown in green)

The two sentences are concatenated with the hidden state in sentence decoder $h_{doc}$ and document representation $v_{doc}$
Text Compression: Continued

- **Compression Classifier**
  - Concatenated representations fed into a feedforward neural network
  - Predicts whether to keep or delete the compression

- **Heuristic Deduplication**
  - Performed after model prediction and compression
  - Compress deletable chunks with redundant information
    - Remove compressions whose unigrams are covered elsewhere in the document
Training

- Model makes sentence extraction and compression
- Compares these decision to gold-standard labels
- Oracles are constructed
- ROUGE score used
  - Recall-Oriented Understudy for Gisting Evaluation
  - A set of metrics for evaluating automatic summarization of texts as well as machine translations
Sentence Extractive Oracle

- Select sentences to add using beam search
  - Beam Search, similar to MMR
    - A heuristic search algorithm that explores a graph by expanding the most promising node in a limited set
- Compute a heuristic cost equal to ROUGE score of a sentence with respect to reference summary
- Pruning done based on ROUGE score of the combination of sentences currently selected and sorted in descending order
- Beam search returns a beam of B different sentence combinations
Oracle Compression Labels

- Provide binary labels to the compression decisions in extracted sentences
- For simplicity, context is ignored for labeling
- Stemming used to remove stop words
- Compare ROUGE score of sentence with and without this phrase
- If ROUGE score increases, the compression should be applied
Grammaticality

Evaluate grammaticality of compressed summaries

- Amazon Mechanical Turk
  - Noisy and inconsistent
- Grammarly software
- Manual Analysis
**Experimental Results**

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead (Ours)</td>
<td>29.1</td>
<td>11.1</td>
<td>25.8</td>
</tr>
<tr>
<td>Refresh* (Narayan et al., 2018)</td>
<td>30.3</td>
<td>11.6</td>
<td>26.9</td>
</tr>
<tr>
<td>LatSum* (Zhang et al., 2018)</td>
<td>28.8</td>
<td>11.5</td>
<td>25.4</td>
</tr>
<tr>
<td>BanditSum (Dong et al., 2018)</td>
<td><strong>30.7</strong></td>
<td><strong>11.6</strong></td>
<td><strong>27.4</strong></td>
</tr>
<tr>
<td>LEADDEDUP</td>
<td>29.7</td>
<td>10.9</td>
<td>26.2</td>
</tr>
<tr>
<td>LEADCOMP</td>
<td>30.6</td>
<td>10.8</td>
<td>27.2</td>
</tr>
<tr>
<td>EXTRACTION</td>
<td>30.3</td>
<td>11.0</td>
<td>26.5</td>
</tr>
<tr>
<td>EXT-LSTMDEL</td>
<td>30.6</td>
<td>11.9</td>
<td>27.1</td>
</tr>
<tr>
<td>JECS</td>
<td><strong>32.7</strong></td>
<td><strong>12.2</strong></td>
<td><strong>29.0</strong></td>
</tr>
</tbody>
</table>

Table 3: Experimental results on the test sets of CNN. * indicates models evaluated with our own ROUGE metrics. Our model outperforms our extractive model and lead-based baselines, as well as prior work.

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead (Ours)</td>
<td>40.3</td>
<td>17.6</td>
<td>36.4</td>
</tr>
<tr>
<td>Refresh* (Narayan et al., 2018)</td>
<td>40.0</td>
<td>18.1</td>
<td>36.6</td>
</tr>
<tr>
<td>NeuSum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LatSum* (Zhang et al., 2018)</td>
<td>41.0</td>
<td>18.8</td>
<td>37.4</td>
</tr>
<tr>
<td>LatSum w/ Compression</td>
<td>36.7</td>
<td>15.4</td>
<td>34.3</td>
</tr>
<tr>
<td>BanditSum</td>
<td>41.5</td>
<td>18.7</td>
<td>37.6</td>
</tr>
<tr>
<td>CBDec (Jiang and Bansal, 2018)</td>
<td>40.7</td>
<td>17.9</td>
<td>37.1</td>
</tr>
<tr>
<td>FARS (Chen and Bansal, 2018)</td>
<td>40.9</td>
<td>17.8</td>
<td><strong>38.5</strong></td>
</tr>
<tr>
<td>LEADDEDUP</td>
<td>40.5</td>
<td>17.4</td>
<td>36.5</td>
</tr>
<tr>
<td>LEADCOMP</td>
<td>40.8</td>
<td>17.4</td>
<td>36.8</td>
</tr>
<tr>
<td>EXTRACTION</td>
<td>40.7</td>
<td>18.0</td>
<td>36.8</td>
</tr>
<tr>
<td>JECS</td>
<td><strong>41.7</strong></td>
<td><strong>18.5</strong></td>
<td><strong>37.9</strong></td>
</tr>
</tbody>
</table>

Table 4: Experimental results on the test sets of CNNDM. The portion of CNN is roughly one of tenth of DM. Gains are more pronounced on CNN because this dataset features shorter, more compressed references.
Questions

Thank you for patiently listening to my talk