A Neural Attention Model for Abstractive Sentence Summarization

By: Alexander Rush, Sumit Chopra, Jason Weston
1. Task
2. Motivation
3. Model
   a. Decoder
   b. Encoder
4. Training
5. Generating Summaries
6. Extension
7. Dataset
8. Baselines
9. Results
10. Examples
Task

- Abstractive Text Summarization: Produce a condensed representation of the input text containing the core meaning of the original

- Different from Extractive Text Summarization and Sentence Compression

- In Abstractive Text Summarization the summary is entirely generated from the bottom up
Motivation

● Human summarizers apply different techniques then crop/combine
  ○ They paraphrase, generalize, and reorder sequences

● Data-driven methods applied to Neural Machine Translation have shown great success

● The encoder and generation models are trained jointly on document-summary pairs, allowing the system to scale to large data-sets
Model: Target

- With a vocab of size $V$ and a fixed output length of size $N$
- Then let $Y \subseteq (\{0, 1\}^V, \ldots, \{0, 1\}^V)$ describe all possible summaries
- The optimal sentence can be described as:
  \[
  \arg\max_{y \in Y} s(x, y) \approx \sum_{i=0}^{N-1} g(y_{i+1}, x, y_c)
  \]
- The log probability of a summary given the input is:
  \[
  \log p(y|x; \theta) \approx \sum_{i=0}^{N-1} \log p(y_{i+1}|x, y_c; \theta)
  \]
Model: Decoder

- Their language model is a modified version of the language model in Bengio et al. (2003)

\[
p(y_{i+1}|y_c, x; \theta) \propto \exp(Vh + W_{\text{enc}}(x, y_c)),
\]

\[
\tilde{y}_c = [E_{y_{i-C+1}}, \ldots, E_{y_i}],
\]

\[
h = \tanh(U\tilde{y}_c).
\]

\[\Theta = (E, U, V, W)\]

\[
E \in \mathbb{R}^{D \times V} \\
V \in \mathbb{R}^{V \times H} \\
U \in \mathbb{R}^{(CD) \times H} \\
W \in \mathbb{R}^{V \times H}
\]

- Where \(D\) is the word embedding size, \(H\) is the dimension of hidden layer \(h\)

Figure 3: (a) A network diagram for the NNLM decoder with additional encoder element. (b) A network diagram for the attention-based encoder \(\text{enc}_3\).
Model: Encoder

- The authors explore the use of 3 encoders in their architecture

### Bag-of-Words Encoder

\[
\text{enc}_1(x, y_c) = p^\top \tilde{x},
\]

\[
p = [1/M, \ldots, 1/M],
\]

\[
\tilde{x} = [F_x_1, \ldots, F_x_M].
\]

- \( F \in \mathbb{R}^{H \times V} \)
- \( p \in [0, 1]^M \)
- Where \( p \) is a uniform distribution over the input words

### Convolutional Encoder

\[
\forall j, \text{enc}_2(x, y_c)_j = \max_i \tilde{x}_{i,j}^L,
\]

\[
\forall i, l \in \{1, \ldots, L\}, \quad \tilde{x}_{i,j}^l = \tanh(\max\{\tilde{x}_{2i-1}^l, \tilde{x}_{2i}^l\})
\]

\[
\forall i, l \in \{1, \ldots, L\}, \quad \tilde{x}_i^l = Q^{i-1}\tilde{x}_{[i-Q, \ldots, i+Q]},
\]

\[
\tilde{x}_i^0 = [F_x_1, \ldots, F_x_M].
\]

- Where \( F \) is a word embedding matrix
- And \( Q^{L \times H \times 2Q+1} \) is a set of filters for each layer \( \{1, \ldots, L\} \)
Model: Encoder

Attention-based Encoder

\[
\text{enc}_3(x, y_c) = p^\top \tilde{x},
\]

\[
p \propto \exp(\tilde{x} P \tilde{y}_c'),
\]

\[
\tilde{x} = [Fx_1, \ldots, Fx_M],
\]

\[
\tilde{y}_c' = [Gy_{i-C+1}, \ldots, Gy_i],
\]

\[
\forall i \quad \tilde{x}_i = \sum_{q=i-Q}^{i+Q} \tilde{x}_i / Q.
\]

- Where \( G \in \mathbb{R}^{D \times V} \) is an embedding of the context.
- And \( P \in \mathbb{R}^{H \times (CD)} \) is a new weight matrix, mapping between the context and the input.
- This is effectively BoW with a learned attention matrix.
If we have $J$ pairs of input and ground truth summary pairs then the objective function for training can be described as:

$$
NLL(\theta) = - \sum_{j=1}^{J} \log p(y^{(j)}|x^{(j)}; \theta),
$$

$$
= - \sum_{j=1}^{J} \sum_{i=1}^{N-1} \log p(y_{i+1}^{(j)}|x^{(j)}, y_c; \theta).
$$

They minimize this function using mini-batch stochastic gradient
Generating Summaries

- Calculating the most probable sequence is expensive
- Authors use beam-search to limit options to best $K$ options at each step
- This generates summaries left-to-right
- The algorithm has complexity $O(KNV)$

Algorithm 1: Beam Search

**Input:** Parameters $\theta$, beam size $K$, input $x$

**Output:** Approx. $K$-best summaries

1. $\pi[0] \leftarrow \{e\}$
2. $S = V$ if abstractive else $\{x_i \mid \forall i\}$

   for $i = 0$ to $N - 1$
   
   - Generate Hypotheses
     
     $N \leftarrow \{[y, y_{i+1}] \mid y \in \pi[i], y_{i+1} \in S\}$
   
   - Hypothesis Recombination
     
     $H \leftarrow \{y \in N \mid s(y, x) > s(y', x) \quad \forall y' \in N \text{ s.t. } y_c = y_c'\}$
   
   - Filter K-Max
     
     $\pi[i + 1] \leftarrow \text{K-arg max}_{y \in H} g(y_{i+1}, y_c, x) + s(y, x)$

end for

return $\pi[N]$
Extension: Extractive Tuning

- The authors find the model does not use extractive word matches
  - i.e. recovering proper nouns from the input
- To solve this issue the authors use five added scores weighted on a learned weight vector
- $\alpha$ is a 5-dimensional weight vector
- After training the main model parameters are frozen and then $\alpha$ is tuned

$$s(y, x) = \sum_{i=0}^{N-1} \alpha^T f(y_{i+1}, x, y_c)$$

$$f(y_{i+1}, x, y_c) = [\log p(y_{i+1} | x, y_c; \theta),$$

$$1\{\exists j. y_{i+1} = x_j\},$$

$$1\{\exists j. y_{i+1-k} = x_{j-k} \forall k \in \{0, 1\}\},$$

$$1\{\exists j. y_{i+1-k} = x_{j-k} \forall k \in \{0, 1, 2\}\},$$

$$1\{\exists k > j. y_i = x_k, y_{i+1} = x_j\}].$$
Dataset

- Abstractive Sentence Summarization is focused on headline creation
- The model is trained on the Gigaword dataset (4 Million)
  - Headlines are paired with the first sentence for summary-input pairs
- DUC-2003 and DUC-2004 contests are used for evaluation
  - News articles from NYT and AP with corresponding human created summaries
  - 500 articles
- Scored using recall-oriented ROUGE metric
  - ROUGE-1 (unigram)
  - ROUGE-2 (bigram)
  - ROUGE-L (longest-common-substring)
- They additionally report evaluation scores on a subset of Gigaword
Baselines

- Prefix: Return the first 75 characters of the input as a headline
- Topiary: Winning system on DUC shared task
  - A merged compression and topic detection system
- Woodsend et al. (2010) also report scores on the DUC set
- Human Annotations: Average of 4 different human summarizers
- COMPRESS: A sentence compression model trained on the same data
- IR baseline: Retrieves the closest headline from the training set
- Moses+: Phrase-based statistical machine translation system
### Results

<table>
<thead>
<tr>
<th>Model</th>
<th>ROUGE-1</th>
<th>DUC-2004</th>
<th>ROUGE-L</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>Gigaword</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
<th>ROUGE-L</th>
<th>Ext. %</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR</td>
<td>11.06</td>
<td>1.67</td>
<td>9.67</td>
<td>16.91</td>
<td>5.55</td>
<td>15.58</td>
<td>29.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREFIX</td>
<td>22.43</td>
<td>6.49</td>
<td>19.65</td>
<td>23.14</td>
<td>8.25</td>
<td>21.73</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPRESS</td>
<td>19.77</td>
<td>4.02</td>
<td>17.30</td>
<td>19.63</td>
<td>5.13</td>
<td>18.28</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W&amp;L</td>
<td>22</td>
<td>6</td>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOPIARY</td>
<td>25.12</td>
<td>6.46</td>
<td>20.12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOSES+</td>
<td>26.50</td>
<td>8.13</td>
<td>22.85</td>
<td>28.77</td>
<td>12.10</td>
<td>26.44</td>
<td>70.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABS</td>
<td>26.55</td>
<td>7.06</td>
<td>22.05</td>
<td>30.88</td>
<td>12.22</td>
<td>27.77</td>
<td>85.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABS+</td>
<td>28.18</td>
<td>8.49</td>
<td>23.81</td>
<td>31.00</td>
<td>12.65</td>
<td>28.34</td>
<td>91.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFERENCE</td>
<td>29.21</td>
<td>8.38</td>
<td>24.46</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>45.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The percentage of tokens in the summary that also appear in the input are Ext %
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Encoder</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>KN-Smoothed 5-Gram</td>
<td>none</td>
<td>183.2</td>
</tr>
<tr>
<td>Feed-Forward NNLM</td>
<td>none</td>
<td>145.9</td>
</tr>
<tr>
<td>Bag-of-Word</td>
<td>$\text{enc}_1$</td>
<td>43.6</td>
</tr>
<tr>
<td>Convolutional (TDNN)</td>
<td>$\text{enc}_2$</td>
<td>35.9</td>
</tr>
<tr>
<td>Attention-Based (ABS)</td>
<td>$\text{enc}_3$</td>
<td>27.1</td>
</tr>
</tbody>
</table>

Perplexity of different encoder schemes on the Gigaword evaluation set with a context window of 5.

ROUGE scores on DUC 2003 for various inference methods. Ext is a purely extractive version of the system.
Examples

I(4): australian foreign minister stephen smith sunday congratulated new zealand ’s new prime minister-elect john key as he praised ousted leader helen clark as a “gutsy” and respected politician.

G: time caught up with nz ’s gutsy clark says australian fm
A: australian foreign minister congratulates new nz pm after election
A+: australian foreign minister congratulates smith new zealand as leader
Examples

I(1): a detained iranian-american academic accused of acting against national security has been released from a tehran prison after a hefty bail was posted, a to p judiciary official said tuesday.

G: iranian-american academic held in tehran released on bail
A: detained iranian-american academic released from jail after posting bail
A+: detained iranian-american academic released from prison after hefty bail
Comments & Questions
ROUGE

- ROUGE-1: refers to the overlap of 1-gram (each word) between the system and reference summaries.
- ROUGE-2: refers to the overlap of bigrams between the system and reference summaries.
- ROUGE-L: takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.
Data Cleaning

Items were pruned using the following categories:

1. Are there no non-stop-words in common?
2. Does the title contain a byline or other extraneous editing marks?
3. Does the title have a question mark or colon?