Get To the Point: Summarization with Pointer-Generator Networks

Abigail See (2017)

- Pros
    1. Calculate an explicit switch probability instead of a shared soft–max function, which enables to tune the probability of all generated/copy words at once.
    2. Recycle the attention distribution to serve as the copy distribution
    3. Coverage: a simpler approach – summing the attention distributions to obtain the coverage vector.
- Cons
    1. It doesn't give a complete explanation about comparison with extractive systems. The author sets the reason to the nature of the task and the ROUGE metric. (It seems that extractive systems tend to obtain higher ROUGE scores even not significantly exceed the-3 baseline.)
    2. It doesn't produce novel n-grams. But the baseline model produces more novel n-grams.
    3. Don't have pre-trained the word embedding. So it has a good performance in this summaries follow the X beat Y, similar with the training data contains many sports stories. In general however, this model does not routinely produce summaries. (Pre-train or Self-train? Depends on the variety of topics)
- Future work

    Encouraging the pointer-generator model to write more abstractive, while retaining the accuracy advantages of the pointer module.

Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond

Ramesh Nallapati (2016)

- Pros
    1. Capturing keywords with Feature-rich-encoder: One embedding vector each for POS, NER tags and discretized TF and IDF values, concatenated together with word-based embedding.
    2. Large Vocabulary Trick: The decoder-vocabulary of each mini-batch is restricted to words in the source documents of that batch. The most frequent words in the target dictionary are added until vocabulary reaches the fixed size.
    3. Capturing Hierarchical Document Structure with Hierarchical Attentions: One at the word level and the other at the sentence level. The word-level attention is further re-weighted by the corresponding sentence-level attention.

$$P^a(j) = \frac{P_w^a(j)P_s^a(s(j))}{\sum_{k=1}^{N_d} P_w^a(k)P_s^a(s(k))},$$

- Cons
    1. Despite using pointers, the performance improvement of the overall model is not significant. The author believe it was because the impact of tail distribution of rare words (So does it means the pointer contents satisfy grammatical instead of sensical?)

2. This work train pointer components to activate only for out-of-vocabulary words or named entities instead of freely learning.
3. Maybe it's better to operate directly on the original CNN/Daily Mail text instead of pre-processing the dataset by anonymizing it.

- Future work
  CNN/Daily Mail dataset with long documents, and ordered multi-sentence summaries will help us to build more novel models to generate summaries consisting of multiple sentences.

Abstractive Sentence Summarization with Attentive Recurrent Neural Networks
Sumit Chopra (2016 ACL)

- Pros
  1. Recurrent Decoder (conditional RNN):

Our Elman RNN takes the following form (Elman, 1990):

$$\begin{aligned} h_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\ P_t &= \rho(W_4 h_t + W_5 c_t), \end{aligned}$$

RAS-Elman

$$\begin{aligned} i_t &= \sigma(W_1 y_{t-1} + W_2 h_{t-1} + W_3 c_t) \\ i'_t &= \tanh(W_4 y_{t-1} + W_5 h_{t-1} + W_6 c_t) \\ f_t &= \sigma(W_7 y_{t-1} + W_8 h_{t-1} + W_9 c_t) \\ o_t &= \sigma(W_{10} y_{t-1} + W_{11} h_{t-1} + W_{12} c_t) \\ m_t &= m_{t-1} \odot f_t + i_t \odot i'_t \\ h_t &= m_t \odot o_t \\ P_t &= \rho(W_{13} h_t + W_{14} c_t). \end{aligned}$$

RAS-LSTM

  2. Attentive Encoder (encode the position information): Each word X_i in the input sequence is associated with one aggregate embedding vector Z_i. The vector Z_i can be seen as a representation of the word which captures the position in which it occurs in the sentence and also the context.

$$z_{ik} = \sum_{h=-q/2}^{q/2} a_{i+h} \cdot b^k_{q/2+h},$$

  3. RAS-Elman is simpler than the NMT model at multiple levels but achieve the similar performance. (Haven't present the comparison details.)
- Cons
  1. The fluent but nonsensical summaries still occurs.
  2. Once RAS model mistakes the content of a relative clause for the main, it lead to a summary with the opposite meaning. (Position information part is still a challenges)

3. Maybe we can find a way to add the coverage information in Attentive Encoder to prevent the repetition/nonsensical problem.

- Future work
  The position information method is really helpful to generate the multiple sentences.