Romain Paulus. A Deep Reinforced Model for Abstractive Summarization [J]. arXiv preprint arXiv: 1705.04304, 2017.

1. background

Training a model that can generate long, coherent, and meaningful summaries remain s an open research problem.

2. problem/hypothesis

Automatic summarization models can work in one of two ways: by extraction or by abs traction.

- Extractive models : copy-and-paste, lack in flexibility, robust
- Abstractive models : much harder , produce fluent and coherent summaries

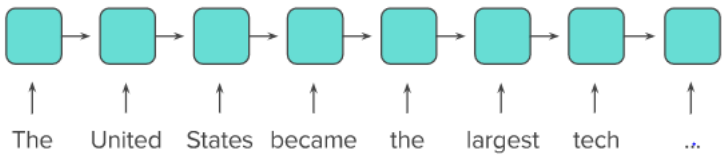
In this work, we tackle these issues and design a more robust and coherent abstractiv e summarization model.

3. solution/argumentation

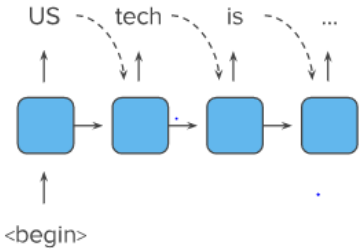
In order to make summarization successful, we introduce two separate improvements

- a more contextual word generation model
- a new way of training summarization models via reinforcement learning (RL).
-

a. Reading and Generating Text with Encoder-decoder Models

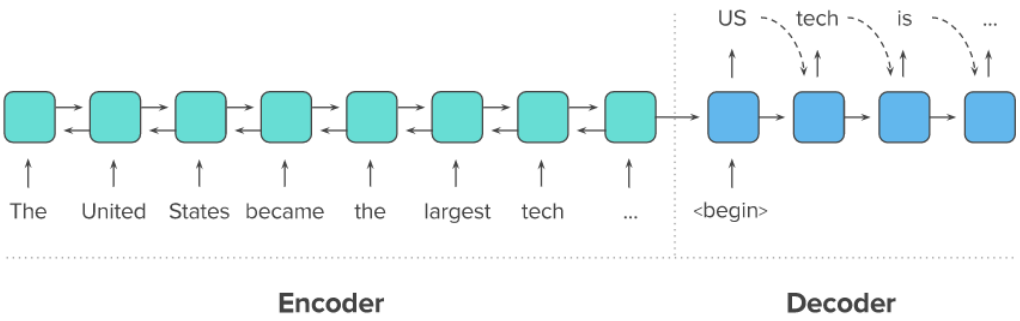


A recurrent neural network reads an input sentence by applying the same function (in green) on individual words.



RNNs can generate output sequences, and re-use the output word as the input of the next function.

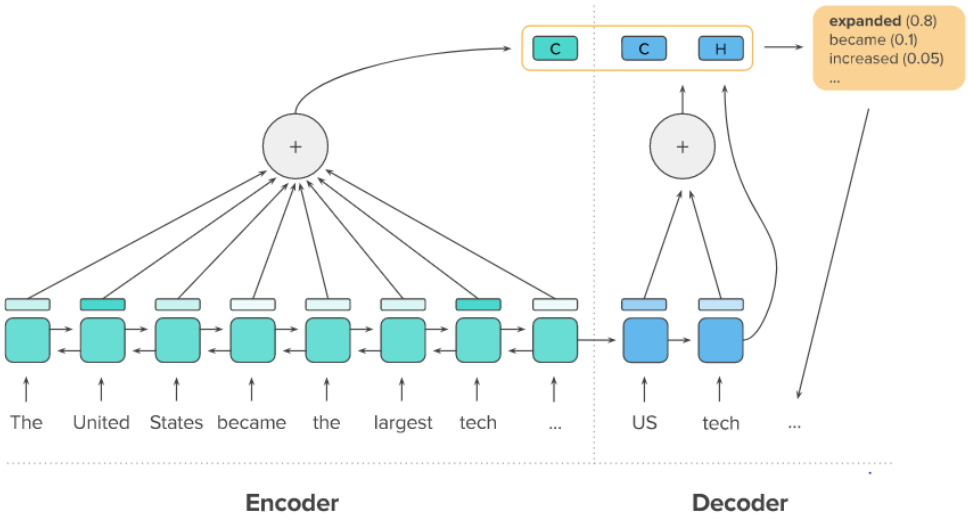
The input (reading) and output (generating) RNNs can be combined in a joint model where the final hidden state of the input RNN is used as the initial hidden state of the output RNN. In addition, we replace the traditional encoder RNN by a bidirectional encoder



Encoder-decoder RNN models can be used to solve sequence-to-sequence tasks in natural language such as summarization.

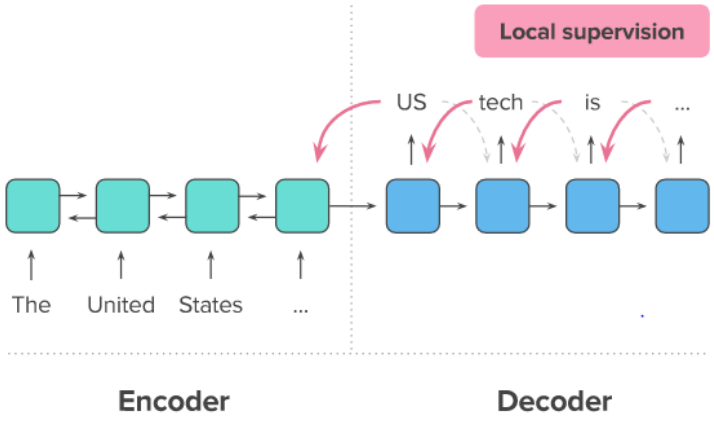
b. A new Attention and Decoding Mechanism

- Temporal attention: the decoder can incorporate contextual information about diff erent parts of the input with an attention function.
- Intra-decoder attention: we allow it to look back at the previous hidden states fro m the decoder.



Two context vectors (marked "C") are computed from attending over the encoder hidden states and decoder hidden states. Using these two contexts and the current decoder hidden state ("H"), a new word is generated (on the right) and added to the output sequence.

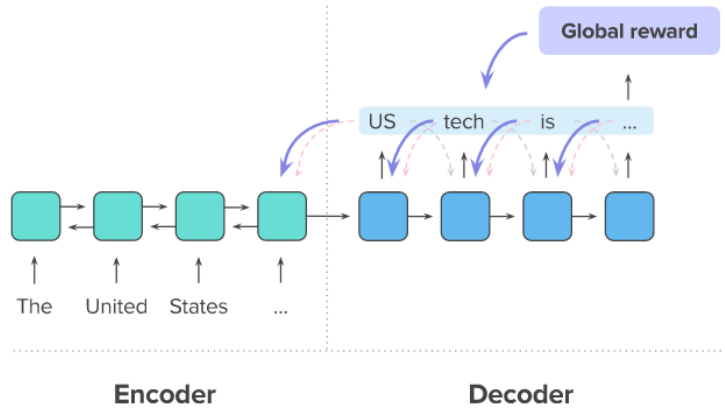
c. Train Model : Supervised Learning vs. Reinforcement Learning



Model training with supervised learning. Each generated word gets a training supervision signal, calculated by comparing it against the ground truth summary word at the same position.

In reinforcement learning, the model doesn't have a local supervision signal for every predicted word, but instead is trained with a reward signal that depends on the entire output and the reference summary.

d. Evaluate Summarization

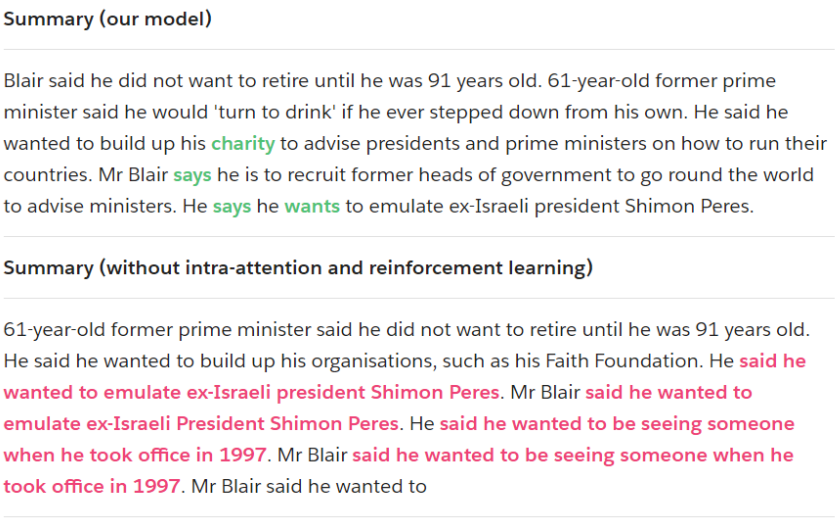


Combination of supervised learning (in red) and reinforcement learning (in purple), showing how our model can learn both local and global rewards and optimize both for readability and overall ROUGE score.

Even though our pure RL model has higher ROUGE scores, our supervised+RL model has a higher readability.

4. experimental test/conclusion

- More examples of summaries generated by ***our model***, compared against summaries ***written by humans*** for the same article.
- In order to illustrate the impact of our main contributions on text summarization, Figure shows how the output of our model diverges if we remove intra-attention and RL training.



Example summaries generated by our model, with and without our main contributions. New words that are not present in the original article are shown in green. Repeated phrases in the summaries are shown in red.

Excellent trick

- Sharing decoder weights (converges faster; improve the token generation function)

Future work

- An ideal automatic evaluation metric (We observed that our models with the highest RO UGE scores also generated barely-readable summaries.)
- The new words generated in the abstract are not always critical(Such as says, wants in t he former figure)

Some doubts

- The reason to use bidirectional encoder in seq2seq improvement。
- How to define the reward signal?
- Why ROUGE has such problem and how to improve it in this evaluation?