A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents

Abstract

Our approach consists of a new hierarchical encoder that models the discourse structure of a document, and an attentive discourse-aware decoder to generate the summary.

I. Introduction

Our decoder attends to different discourse sections and allows the model to more accurately represent important information from the source resulting in a better context vector.

We also introduce two large-scale datasets of long and structured scientific papers obtained from arXiv and PubMed.

2. Background

Attentive Decoding

The attention mechanism maps the decoder state $h_{t-1}^{(d)}$ and the encoder states $h_i^{(e)}$ to context vector c_t. Incorporating this context vector at each decoding timestep (attentive decoding) is proven effective in seq2seq models

$$c_t = \sum_{i=1}^{N} \alpha_i^{(t)} h_i^{(e)}$$

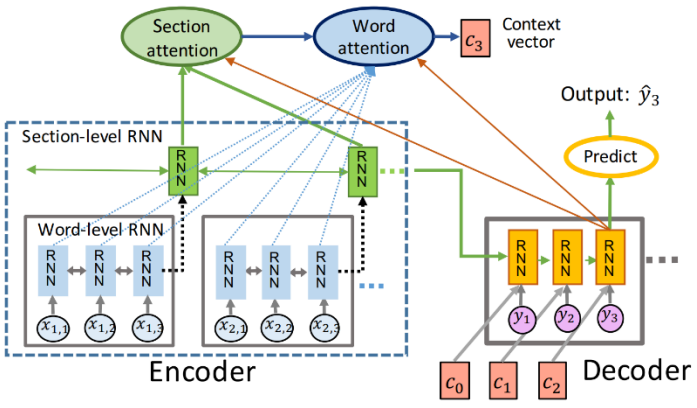
where $\alpha_i^{(t)}$ are the attention weights calculated as follow

$$\alpha_i^{(t)} = \text{softmax}\left(\text{score}\left(h_i^{(e)}, h_{t-1}^{(d)}\right)\right)$$

$$\text{score}\left(h_i^{(e)}, h_{t-1}^{(d)}\right) = v_a^T \tanh\left(\text{linear}\left(h_i^{(e)}, h_{t-1}^{(d)}\right)\right)$$

$$\text{linear}(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$

3. Model



Hierarchical Encoder

We first encoder each discourse section and then encoder the document

$$d = RNN_{doc}\left(\left\{h_1^{(s)},\ \ldots,\ h_N^{(s)}\right\}\right)$$

$$h_j^{(s)} = RNN_{sec}\left(x_{(j,1)},\ \ldots,\ x_{(j,M)}\right)$$

where N is number of sections and M is the maximum section length.

The parameters of RNN_{sec} are shared for all the discourse sections.

We use a single layer bidirectional LSTM for both RNN_{doc} and RNN_{sec}

$$h = \text{relu}\left(W\left[\vec{h},\ \text{cev}(h)\right] + b\right)$$

Discourse-Aware Decoder

At each decoding timestep, in addition to the words in the document, we also attend to the relevant discourse section. Then we use the discourse-related information to modify the word-level attention function.

$$c_t = \sum_{j=1}^{N}\sum_{i=1}^{M} \alpha_{(j,i)}^{(t)}\, h_{(j,i)}^{(e)}$$

$$\alpha_{(j,i)}^{(t)} = \text{softmax}\left(\beta_j^{(t)}.\text{score}\left(h_{(j,i)}^{(e)},\ h_{t-1}^{(d)}\right)\right)$$

$$\beta_j^{(t)} = \text{softmax}\left(\text{score}\left(h_j^{(s)},\ h_{t-1}^{(d)}\right)\right)$$

At each timestep t , the decoder state $h_t^{(d)}$ and the context vector c_t are used to estimate the probability of next word y_t

$$p(y_t|y_{1:t-1}) = \text{softmax}\left(V^T.\text{linear}\left(h_t^{(d)}, c_t\right)\right)$$

where V is a vocabulary weight matrix.

Copying from source

Address the problem of unknown token prediction by allowing the model to occasionally copy words directly from source instead of generating a new token.

We add an additional binary variable z_t to the decoder, indicating generating a word from vocabulary (z_t = 0) or copying a word from the source (z_t = 1). The probability is learnt during training according to the following equation

$$p(z_t = 1|y_{1:t-1}) = \sigma\left(\text{linear}\left(h_t^{(d)}, c_t, x_t'\right)\right)$$

where x_t' is decoder input at timestep t.

Then the next word y_t is generated according to

$$p(y_t|y_{1:t-1}) = \sum_z p(y_t, z_t = z|y_{1:t-1}) \, ; z = \{0,1\}$$

The joint probability is decomposed as

$$p(y_t, z_t = z) = p_c(y_t|y_{1:t-1})p(z_t = z|y_{1:t-1}); z = 1$$

$$p(y_t, z_t = z) = p_g(y_t|y_{1:t-1})p(z_t = z|y_{1:t-1}); z = 0$$

where p_g is the probability of generating and p_c is probability of copying a word from the source

$$p_c(y_t = x_l|y_{1:t-1}) = \sum_{(j,i):x_{(j,i)}=x_l} \alpha_{(j,i)}^{(t)}$$

Decoding Coverage

In long sequences, the neural generation models tend to repeat phrases where the softmax layer predicts the same phrase multiple times over multiple timesteps.

We track attention coverage to avoid repeatedly attending to the same steps with a coverage vector

$$cov_{(j,i)}^{(t)} = \sum_{k=0}^{t-1} \alpha_{(j,i)}^{(k)}$$

We incorporate the decoder coverage as an additional input to the attention function

$$\alpha_{(j,i)}^{(t)} = \text{softmax}\left(\beta_j^{(t)}.\text{score}\left(h_{(j,i)}^{(e)}, cov_{(j,i)}^{(t)}, h_{t-1}^{(d)}\right)\right)$$

4. Related Work

5. Data

6. Experiments

7. Conclusion and Future Work

May 18, 2023

Chu Dinh Duc