A Novel Approach for Text Summarization

Kaushal Prasad Hial, Kalyan Das *

Sambalpur University Institute of Information Technology

Jyoti Vihar, Burla, Odisha, India.

kalyandas1983@gmail.com

Abstract – In recent years neural network based sequence to sequence (seq2seq) model have significantly increased the quality of the automatic text summarization. This is an abstractive way of summarization in which the meaning system forms new sentences instead of simply selecting and copying from the input text. Many new techniques have got added to this to make summary better. But still, it has a few shortcomings. In this paper, we have focused on one of those, which is to focus on a certain part of the input text to produce a summary. In this work, we have introduced a novel approach which first highlights some of the important key phrases and taking those key phrase along with the input text produces an output summary.

Index Terms – seq2seq model, neural network, text summarization.

I. Introduction

Text summarization is the task of condensing of the large input text keeping most of the relevant information intact. The goal is to produce a human-like text summary. Earlier most successful summarization systems were based on extractive approaches that pull out and copy it together with other portions of the text to produce a condensed version. In contrast, abstractive summarization systems try to reproduce the shorter form of the text, while doing so system generates its own interpretation which may contain different parts than the original text [1] [2]. Some of the recent new models are abstraction based. Seq2seq model [3] [4] [5] is one of those state-of-the-art mechanisms. Although this model has produced promising results sill we are not close enough to reach the goal. So we have come up with a model by making some changes to those previously known techniques in order to achieve the goal of producing better summarization.

In our work, we have taken seq2seq model as our base. In the architecture of regular seq2seq model there are two parts; one is a neural network based encoder which encodes the variable length input and produces the fixed sized vector representation of the input. Second is decoder which takes the encoder output and produces the desired output.

In case of humans, we read the text and highlight some of the important parts of the input in our mind and at last, we combine all of that highlighted part and makes a summary out of that.

Keeping this intuition in our mind we have come up with a model which have three parts; first is regular encoder; second is highlighter which highlights some of the key parts of the input; third is the decoder which takes input from the encoder and the highlighter, then tries to focus more on that part of the encoder output which have been highlighted by the highlighter. This decoder also consists of Attention mechanism Pointer Generator Network and Coverage model.

II. MODEL

In most of the earlier famous known models, we can find some form of attention mechanism [6] which is a neural network based mechanism to give more focus to some of the important parts of the input. As there is no mechanism to tell which parts are important, this mechanism sometimes fails to provide proper attention to the proper part of the input. Now with the help of the highlighter, we can explicitly tell the attention mechanism to give more attention to that part of the encoder output which is being considered as important by the Highlighter as given in fig. 1. Highlighter is basically seq2seq neural biased decoder which processes key phrase in the form of a set of encoder hidden state. Highlighter selects all those hidden states of an encoder which are formed in alignment with the time steps of encoding of the highlighter selected key phrase. Now

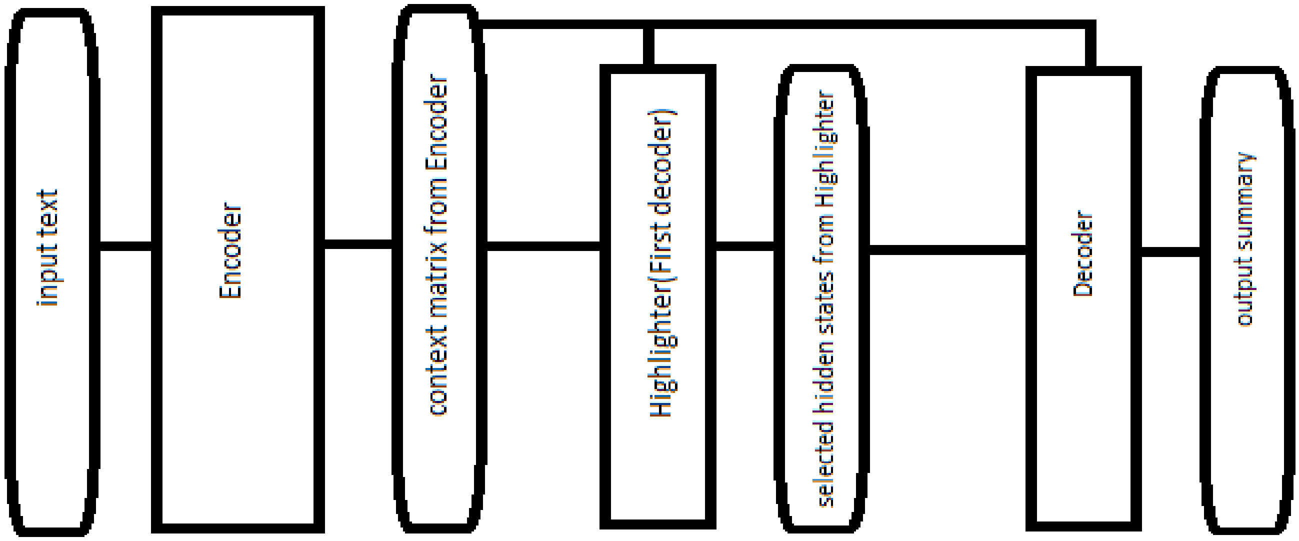


Fig. 1. The proposed model.

with enough amount of data, we can train our highlighter to find important parts. From left, first, there is an encoder which encodes the input text. Then our highlighter which selects all the important hidden states which are having important information.s At last there is decoder which decoded to the desired summary

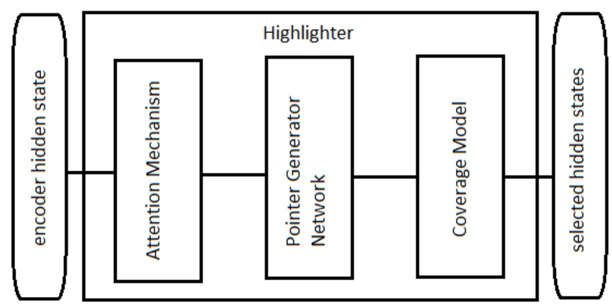
We will talk about highlighter in the latter of this section. For now, we will assume that highlighter has produced output in the form of a set of encoder hidden state, h_li ∈ H_l (set of highlighter selected hidden states). Now let us talk about encoder first.

A Encoder:

The encoder in a Bidirectional Recurrent Neural Network [7] [8] [9]. For every time step encoder produces an output and a hidden state. The hidden state is a vector representation of what the encoder have read so far. The input to the network may be of variable length but the hidden state is a fixed length vector representation of the input. As the hidden state is of a fixed length then it becomes easy to deal and manipulate that is why we are mainly focusing on the hidden state, not on the actual output of the Recurrent Neural Network Encoders output. So now we can say that the job of the encoder is to convert variable length input into a fixed length representation in this case hidden state.

B Decoder:

The decoder is similar to regular decoders in seq2sqe model with some changes in the attention



mechanism [10] [11] to fit for our work. Along with attention, the decoder also contains the pointer generator network and coverage mechanism [12]. Pointer generator network helps the decoder to deal with out of vocabulary words and coverage model

Fig. 2. Image showing the inner model of Highlighter

restricts decoder to give more than enough importance to a certain part of the input.

Originally as in Bandananu et al. (2016), the attention distribution, at is calculated as:

$$e_i^t = V^T \tanh(W_h h_i + W_s s_t + b_{attn}) \qquad -(1)$$

$$a_i^t = \text{softmax}(e_i^t) \qquad -(2)$$

Where V, W_h, W_s and b_{attn} are learnable parameters.

In our work, we made changes to both the above equation. In order to give more attention to highlighter selected hidden states ($h\_l_i \in H\_L$) we changed the above equation as:

$$e_i^t = \begin{cases} V^T \tanh(W_h h_i + W_s s_t + b_{attn})\ if\ (h_i \in \text{H\_L}) \\ V^T \tanh(W_h h_i + W_s s_t - b_{attn})\ if\ (h_i \notin \text{H\_L}) \end{cases} - (3)$$

Now according to this equation b_{attn} is added to every e_i^t of the hidden state, *hi* if *hi* is one of the elements of the set of hidden states which are selected by the Highlighter and b_{attn} is subtracted if *hi* doesn't belong to the set H_l Secondly we have used approach of memorizing the alignment temporally as mentioned in [13]. This approach is used to deal with two issues: i. repetition of words due to more attention to a part of the text than needed; ii. Fail in focusing on some important part of the text.
We calculate a^t as:

$$\mathrm{b_{t,j}} = \sum_{k=1}^{t-1} \exp(e_{k,j}) - (4)$$

Where $b_{t,j}$ denotes the aggregated alignment until the current time step. Modulated alignment $b_{t,j}$ is calculated as:

$$b_{t,j} = \frac{\exp(e_{t,j})}{b_{t,j}^h} - (5)$$

At t=1,

$$b_{t,j} = \exp\left(e_{t,j}\right) - (6)$$

As there is no previous history.

Finally $a_{t,j}$ becomes:

$$a_{t,j} = \frac{b_{t,j}}{\sum_{i=1}^{L} b_{t,i}} - (7)$$

Along with this, the decoder also consists of pointer generated network, for dealing with OOVs and coverage model c^t

$$c^t = \sum_{t'=0}^{t-1} a^{t'} - (8)$$

Including this, in equation (3) we get:

$$e_i^t = \begin{cases} V^T \tanh(W_h h_i + W_s s_t + W_c c_i^t + b_{attn})\ if\ (h_i \in \text{H\_L}) \\ V^T \tanh(W_h h_i + W_s s_t + W_c c_i^t - b_{attn})\ if\ (h_i \notin \text{H\_L}) \end{cases} - (9)$$

C. Highlighter:

As mentioned earlier, highlighter acts as the first decoder of our model and it gives key phrases in the form of a set of encoder hidden state. It is similar to the decoder part of our model as it is also sequence to sequence model, it also has a neural network based attention mechanism along with the pointer generator model and coverage mechanism.
In our word we first trained the encoder and highlighter with a dataset which is specifically designed for keyword extraction task. Then we trained this along with the decoder part on a training dataset which is specifically designed for text summarization task.

III. CONCLUSION

We propose a model which is an extension of the seq2seq model. Our model contains three parts; 1. First is an encoder, which encodes the given input text into fixed sized vector form; 2. Highlighter, this produces a list of the important hidden state which can be used by the decoder; 3. Decoder gives more attention to the Highlighter produced the hidden state. It also has a temporal attention mechanism. This model can be used for abstractive text summarization.

IV. REFERENCES

[1] Jacob Krantz, Jugal Kalita; Abstractive Summarization Using Attentive Neural Techniques; arXiv:1810.08838v1 [cs.CL] 20 Oct 2018.

[2] Junyang Lin, Xu Sun, Shuming Ma, Qi Su; Global Encoding for Abstractive Summarization; arXiv:1805.03989v2 [cs.CL] 10 Jun 2018.

[3] Ilya Sutskever, Oriol Vinyals, Quoc V. Le; Sequence to Sequence Learning with Neural Networks; arXiv:1409.3215v3 [cs.CL] 14 Dec 2014.

[4] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy; Neural Abstractive Text Summarization with Sequence-to-Sequence Models; arXiv:1812.02303v2 [cs.CL] 7 Dec 2018.

[5] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar G ˘ ulçehre, Bing Xiang; Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond; arXiv:1602.06023v5 [cs.CL] 26 Aug 2016

[6] Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio; Neural Machine Translation By Jointly Learning To Align And Translate; arXiv:1409.0473v7 [cs.CL] 19 May 2016.

[7] Jiajun Zhang, Long Zhou, Yang Zhao, Chengqing Zonga; Synchronous Bidirectional Inference for Neural Sequence Generation; arXiv:1902.08955v1 [cs.CL] 24 Feb 2019.

[8] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, Noam Shazeer: Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks; arXiv:1506.03099v3 [cs.LG] 23 Sep 2015

[9] Mike Schuster and Kuldip K. Paliwal; Bidirectional Recurrent Neural Networks; IEEE Transactions On Signal Processing, Vol. 45, No. 11, November 1997.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin; Attention Is All You Need; arXiv:1706.03762v5 [cs.CL] 6 Dec 2017.

[11] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, Yoshua Bengio; Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.

[12] Abigail See, Peter J. Liu, Christopher D. Manning; Get To The Point: Summarization with Pointer-Generator Networks; arXiv:1704.04368v2 [cs.CL] 25 Apr 2017.

[13] Baskaran Sankaran, Haitao Mi, Yaser Al-Onaizan, Abe Ittycheriah; Temporal Attention Model for Neural Machine Translation; arXiv:1608.02927v1 [cs.CL] 9 Aug 2016.