Abstractive Text Summarization

Prakash Nath Jha - 2018201013

Nitish Srivastava - 2018201012

April 30, 2020

1 Abstract

Neural abstractive text summarization with sequence-to-sequence Seq2seq models have gained a lot of popularity in recent years. Most of these Seq2seq methods mostly differs in network structures, attention mechanism, sentence generation/decoding. However these models suffers from many shortcommings like failure to accurately reproduce the salient infromation of source documents, inability to deal with out of vacabulary (OOV) words, words tends to repeat themselves etc. In the paper[1] we have choosen to study and implement tends to address problems associated with unidirectional nature of Seq2Seq model which fails to incorporate future context during encoding/decoding phase by proposing Bi-Directional encoder and decoder to incorporate both past and future context. To handle words repetitions authors[1] proposes coverage mechanism. Also during inference step authors[1] proposes use of Bi-Directional Beam Search to overcome the problems associated with traditional unidirectional beam search algorithm. The model[1] proposed tends to handle unbalanced output. We have implemented the model suggested in the paper[1] for Google News Dataset[8] .

2 Introduction

In this day and age we have been overwhelmed by limitless data. It has been said that "It is not information overload, it's filter failure.". To tackle this issue, much attention has been paid to Automatic Document Summarization that alleviates this problem by reducing the size of along document to a few sentences or paragraphs. Text summarization is the task of reducing a text, such as a sentence, article, to a shorter version that retains the core ideas of the original text. Before 2014, the majority of work regarding text summarization made use of the extractive approach with relative success. After the development of sequence to sequence models, abstractive summarization (based on seq2seq models) became widespread, competing with extractive models for state-of-the-art results.In this

project we propose to build a RNN(LSTM) based Seq2seq model using attention mechanism and coverage for the task of abstractive text summarization. We also implement bi-directional beam search decoder for inference.

3 Literature survey

For the task of text summarization there are broadly two main approches:

- Extractive methods which build summaries from selecting and compressing input text sequences. Without actually understanding the sentences, it acts as a highlighter for the document.
- Abstractive methods which build summaries more like a human would, by creating new text passages containing words selected from a wider vocabulary.

Most of the abstractive techniques for text summarization differ in one of these three categories:

- Network/Model structure
- Parameter inference
- Decoding generation

Main Strategies for Abstractive Text Summarization

- AMR representation graph
- Deep learning based Seq2seq Model

Nallapati et al [12]. introduced several novel elements to the RNN encoder-decoder architecture to address critical problems in the abstractive text summarization, including using the following feature-rich encoder to capture keywords, a switching generator-pointer to model out-of-vocabulary (OOV) words, and the hierarchical attention to capture hierarchical document structures. The model we have trained generates only a single sentence summary mainly because the dataset we have chosen to train on only has a single sentence text and summary. Multi sentence summary models can be trained on CNN/Daily Mail dataset.

4 Research Methods

- WordEmbeddingLayer
  We experimented with both custom word embedding layer and with word embedding trained on Fasttext model on our dataset. Both word embeddings sizes were 250 dimensional.

$$\overrightarrow{h_t^e} = \overrightarrow{LSTM}(x_t, \overrightarrow{h_{t-1}})$$
$$\overleftarrow{h_t^e} = \overleftarrow{LSTM}(x_t, \overleftarrow{h_{t+1}})$$

Figure 1: The Encoder

- **Encoder**
  In encoder we have a two uni-directional LSTM one for forward flow of context and other for backward flow of context.

- **Decoder**
  In decoder side we again have two uni-directional LSTM similar to the structure of encoder. The output of the forward encoder was fed as input into the backward decoder while the output of the backward encoder was fed into the forward decoder.

$$P(y_t|[y_d]_{d\neq t}) = \overrightarrow{\log p(y_t|Y_{[1:t-1]})} + \overleftarrow{\log p(y_t|Y_{[t+1:T_y]})} \quad (3)$$

Where $\overrightarrow{\log p(y_t|Y_{[1:t-1]})}$ and $\overleftarrow{\log p(y_t|Y_{[t+1:T_y]})}$ are the left-to-right and the right-to-left LSTM decoder model, Eq. 4 and Eq. 5 respectively.

$$\overrightarrow{\log p(y_t|Y_{[1:t-1]})} = \sum_{t=1}^{T_y} \log p\left(y_t|\{y_1, \ldots, y_{t-1}\}, x; \vec{\theta}\right) \quad (4)$$

$$\overleftarrow{\log p(y_t|Y_{[t+1:T_y]})} = \sum_{t=1}^{T_y} \log p\left(y_t\middle|\left\{y_{t+1}, \ldots, y_{T_y}\right\}, x; \overleftarrow{\theta}\right) \quad (5)$$

$$p\left(y_t|\{y_1, \ldots, y_{t-1}\}, x; \vec{\theta}\right) = g(y_{t-1}, \overrightarrow{h_t^d}, c_t) \quad (6)$$

$$\overrightarrow{h_t^d} = LSTM(y_{t-1}, \overrightarrow{h_{t-1}^d}, c_t) \quad (7)$$

$$p\left(y_t\middle|\left\{y_{t+1}, \ldots, y_{T_y}\right\}, x; \overleftarrow{\theta}\right) = g(y_{t+1}, \overleftarrow{h_t^d}, c_t) \quad (8)$$

$$\overleftarrow{h_t^d} = LSTM(y_{t+1}, \overleftarrow{h_{t+1}^d}, c_t) \quad (9)$$

Figure 2: The Decoder

- **Attention Layer**

$$e_{ij} = v^T \tanh(W_h^d h_i^d + W_h^e h_j^e + b_{attn}) \quad (10)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} e_{ik}} \quad (11)$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (12)$$

Figure 3: Attention Layer

5 Dataset

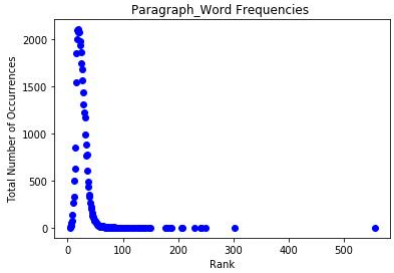

Figure 4: Paragraph_Word_Frequency

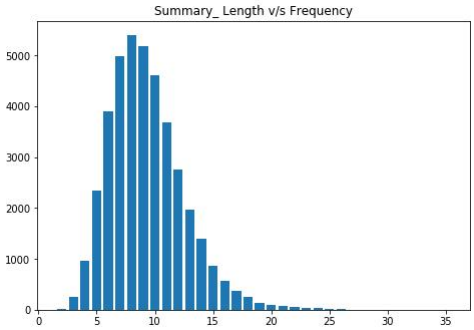

Figure 5: Summary_Word_Frequency

We are using Google sentence compression dataset, which is a large corpus of uncompressed and compressed sentences from news articles. Link: https://github.com/google-research-datasets/sentence-compression

6 Findings and analysis

We have various forms of the basic model proposed in the paper. The main categories are following:

Model	**Mechanism**
Baseline	Unidirectional Enc-Dec + greedy search
Bi_BS	20cmBi-directional Enc-Dec + Attention + Beam Search
Bi_Cov_BS	Bi-directional Enc-Dec + Attention + Coverage + Beam Search
Bi_Cov_BS_FT	Bi-directional Enc-Dec + Attention + Coverage + Beam Search + Fasttext
Bi_Cov_BBS_FT	Bi-directional Enc-Dec + Attention + Coverage + Bi-Directional Beam Search + Fasttex

Model	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
Baseline	18.28	3.654	20.258
Bi_BS	23.596	5.79	23.75
Bi_Cov_BS	24.687	5.78	24.486
Bi_Cov_BS_FT	26.79	6.023	27.78
Bi_Cov_BBS_FT	28.292	9.39	28.50

Table 1: F1 values for ROUGE parameters

From the results we can see that compared to the baseline model the variants of the model proposed in the paper provides significant improvement. We can also observe that ROUGE 2 score is significantly less for all the models and our best model could only get ROUGE 2 score of 9.39%. One of the main reasons behind it could be the fact that out dataset set contains single sentence passage and summaries are also short single sentence summary due to which model get very less oppurtunity to improve its score.

7 Discussion

Due to the dataset choosen has only single sentence as passage and single sentence summary it limits our model capability for producing multi-sentence summary.

8 Limitations

One of the limitations which we can observe is that the performance decreases with increase in length of sentences. Since the model we have trained used Fasttext model for word embedding it makes the model very bulky due to the large size of Fasttext embedding and also slows down training. Due to the dataset choosen has only single sentence as passage and single sentence summary it limits our model capability for producing multi-sentence summary. ROUGE score and cross entropy loss may not go hand in hand and therefore for some cases we observed decreasing cross entropy loss may also lead to decrease in ROUGE score. This makes the evalution more difficult. Relation of ROUGE score with cross entropy loss is a interesting topic for futher study.

9 Future scope

Paper[1] which we refered had a custom word embedding layer and did not make use of any pretrained word embedding layer. But we experimented with both custom layer and Fasttext word embedding layer. We can use more powerful and state of the art word embeddings like ELMo[9] or BERT[10] to give our model better word representation. We can also additionaly have a character embedding

layer which can help in the case of OOV words. To deal with the problem associated with non differentiable scores like ROUGE we can use reinforcement based approches like REINFORCE[11] model. REINFORCE[11] algorithm can make use of any user-defined task specific reward. Finding a strategy to encorage models to generate more novels for summaries is an active area of research.

Code Link: https://github.com/pnjha/Text$_{S}ummarization$

10 References

[1] Kamal Al-Sabahi, Zhang Zuping, Yang Kang Bidirectional Attentional Encoder-Decoder Model and Bidirectional Beam Search for Abstractive Summarization.
Link: https://arxiv.org/pdf/1809.06662v1.pdf

[2] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy, Senior Member, IEEE Neural Abstractive Text Summarization with Sequence-to-Sequence Models: A Survey
Link:https://arxiv.org/pdf/1812.02303.pdf

[3] Bidirectional Beam Search: Forward-Backward Inference in Neural Sequence Models for Fill-in-the-Blank Image Captioning.
Link : https://arxiv.org/abs/1705.08759

[4] When to Finish? Optimal Beam Search for Neural Text Generation (modulo beam size)
Link : https://arxiv.org/abs/1809.00069

[5] Abigail See, Peter J. Liu, Christopher D. Manning, Get To The Point : Summarization with Pointer-Generator Networks
Link: https://arxiv.org/pdf/1704.04368.pdf

[6] Piotr Bojanowski, Edouard Grave, Armand Joulin ,Tomas Mikolov Enriching Word Vectors with Subword Information
Link: https://arxiv.org/pdf/1607.04606.pdf

[7] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space
Link: https://arxi.org/pdf/13v01.3781.pdf

[8] Google News Dataset for Sentence Compression
Link: https://github.com/google-research-datasets/sentence-compression

[9] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner Deep contextualized word representations
Link: https://arxiv.org/pdf/1802.05365.pdf

[10] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova BERT: Pre-t raining of Deep Bidirectional Transformers forLanguage Understanding
Link : https://arxiv.org/pdf/1810.04805.pdf

[11] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," in Reinforcement Learning. Springer, 1992, pp. 5–32.

[12] R. Nallapati, B. Zhou, C. dos Santos, Ç. glar Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," CoNLL 2016, p. 280, 2016.