International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5S3, July 2019

A Pointer Generator Network Model to Automatic Text Summarization and Headline Generation

Anubha Agrawal, Sakshi Saraswat, Hira Javed

Abstract*: : In a world where information is growing rapidly every single day, we need tools to generate summary and headlines from text which is accurate as well as short and precise. In this paper, we have described a method for generating headlines from article. This is done by using hybrid pointer-generator network with attention distribution and coverage mechanism on article which generates abstractive summarization followed by the application of encoder-decoder recurrent neural network with LSTM unit to generate headlines from the summary. Hybrid pointer generator model helps in removing inaccuracy as well as repetitions. We have used CNN / Daily Mail as our dataset.

Index Terms*: LSTM Encoder Decoder Model, Natural Language Processing, Pointer generator network and Coverage Mechanism, Text Summarization.

I. INTRODUCTION

We need summarization of our text to determine essential ideas and consolidate important details. It helps to focus on key words and phrases of text that are worth noting and remembering. The amount of information available online is overwhelming. Text summarization focuses on presenting information effectively and concisely.

According to WordNet(Princeton) summary is defined as "a brief statement that presents the main points in a concise form". Automatic Text Summarization is a process of generating summaries by a computer program.Summarization process involves interpretation, transformation and generation. There are two types of Summarization: extractive and abstractive. In extractive summarization, the automatic system copies the words from the entire text, without modifying the text themselves. It is very similar to highlighting the important sentence in the document. Abstractive summarization involves paraphrasing sections of the source document.

Majority of work is done on extractive summarization in past but now a day researches are mainly focused on abstractive summarization.

For abstractive summarization we have used pointer generator network [3]. First of all, we took documents and apply a pointer generator model [3] to it and then we have generated headlines from it using another model.

Sequence-to-Sequence model can also be used for abstractive summarization but there are two main problems that are associated with it. First is that it produces Out of Vocabulary words and second one is the repetition of words.

In this paper we focus on the issue of long-text summarization because most of the work is done on reducing one shorter paragraph to single line summary which requires higher level of abstraction and we also have a focus on avoiding repetition. Therefore, we use CNN / Daily Mail Dataset which contains news articles and multi sentence summary. And for generating headline of the articles we pass the summarized text to LSTM encoder decoder model.

II. RELATED WORK

Previous works have been done on LSTM encoder decoder model and attention mechanism. Works in [1] and [2] uses LSTM encoder decoder model with attention mechanism. Authors in [1] show that the neural network decides the function of the different neurons and it identifies the input words we have to attend in a simplified attention mechanism. Recent advancement in technology of machine learning like recurrent neural network produces better quality of text summarization. Abstractive summarization is challenging and more effective than extractive summarization [18] [19]. [2] utilizes a local attention-based model which generates each word of the summary conditioned on the input sentence.

In [3] the authors have used attention distribution which is a probability distribution. Basically it is telling where to focus on sequence of generated summary. Pointer generator model is very similar to sequence to sequence model [12]. Output sequence consist of input sequence which is produced by soft attention distribution hybrid approaches for language modeling [14], NMT [15] and summarization [15], [16], [17]. Coverage was originating from Statistical Machine Translation [21] and by NMT. [22],[23] both have used GRU to update a coverage vector. Coverage vector is obtained by summing of attention distribution. So our approach is similar to [24], where coverage mechanism for image capturing has been applied. Authors in [25] have used coverage mechanism for neural summarization on longer text.

III. SYSTEM OVERVIEW

We have taken our dataset from CNN/Daily mails [4] which consists of multi sentence summaries of online news article.

Manuscript published on 30 July 2019.
* Correspondence Author (s)
Anubha Agrawal, Computer Engineering Department, Aligarh Muslim University, Aligarh, India.
Sakshi Saraswat, Computer Engineering Department, Aligarh Muslim University, Aligarh, India.
Hira Javed, Computer Engineering Department, Aligarh Muslim University, Aligarh, India.

We fed the dataset into a pointer generator model. But it produced factual details inaccurately and there was repetition of some words in summary. Owing to this, we have used attention distribution and coverage mechanism in addition to pointer generator model as described in [3]. Now we have got multiple sentence abstractive summary of news articles. In the next step we passed these summaries through LSTM encoder decoder model for generating headlines as described in [1]. We trained LSTM encoder decoder model on CNN/Dailymail stories [4]. We have evaluated our model on our own created dataset which has been obtained from summary generated from pointer generator model.

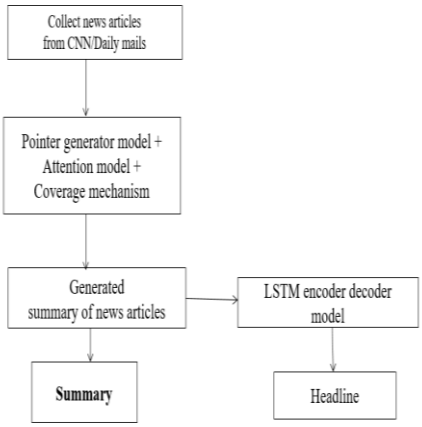

Fig. 1 Block Diagram for Generating Headlines

IV. EXPERIMENT

A. Dataset

We have taken dataset from [4]. We trained our model using CNN/Dailymail stories, specifically using the description of a review as our input data, and the title of a review as our target data. We have transformed the dataset into the binary format. This dataset contains about 300000 articles.

B. Preparing Data

For pointer-generator model, we downloaded and unzipped the story directories from [4] for CNN and Dailymail. We tokenized the data and processed the data into .vocab and .bin file. This script consisted of two directories containing tokenized I version of CNN and Dailymails. Then we split our dataset into 70% training set, 20% evaluation set, and 10% testing set. For LSTM model, we converted our data into lower case. Then replaced contractions form of word with their longer form and then we removed stopwords from it. We will be using pre-trained word vector which will help in improving the performance of the model. We have used set of word embedding called Concept Network Batch which is a pre trained word vector. It is better since it also contains embeddings of Glove.We will be sorting the reviews on their length basis from shortest to longest. Some reviews are not included because of the number of UNK tokens. We have prepared our own standard dataset for evaluation by finding the headlines of the summaries on google.

C. Pointer Generator Model

Encoder RNN read source text word by word and produces a sequence of bidirectional encoder hidden states. Decoder RNN produces output as a sequence of words when Encoder RNN finishes its reading. At each step decoder takes input and previous word of summary to update the decoder hidden states. This is used to calculate the probability distribution of words known as attention distribution. Attention distribution helps the network where to look to produce the next word. Weighted sum of encoder hidden states is produced by attention distribution known as context vector. Now, the context vector and the decoder hidden states are used to produce a probability distribution known as vocabulary. The Decoder chooses the word with largest probability before moving on to the next word.

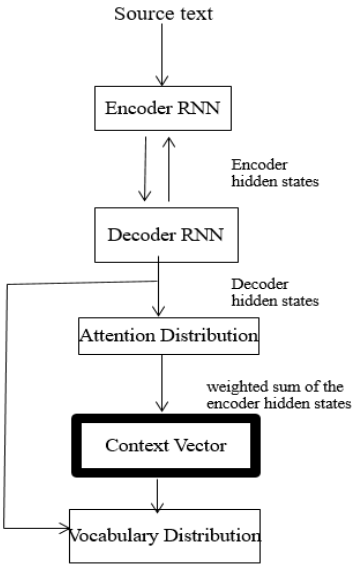

Fig. 2 Block Diagram of Pointer Generator Network

Pointer generator network copy words from source via pointing and use fixed vocabulary to generate new words. We also calculate generation probability that represent a copy of word from source text versus generating a new word from vocabulary. We use attention distribution **a** (for copying of words), vocabulary distribution P_{vocab} (for generating a new word) and generation probability $\wp_{gen}$ (for weight) to calculate final distribution P_{final} .

Link of Article	Actual Summary	Decoded Summary	Actual Headline	Decoded Headline
https://edition.cnn.com/2015/04/01/europe/france-germanwings-plane-crash-main/index.html	marseille prosecutor says `` so far no videos were used in the crash investigation " despite media reports . journalists at bild and paris match are `` very confident " the video clip is real , an editor says . andreas lubitz had informed his lufthansa training school of an episode of severe depression , airline says .	french prosecutor says he was not aware of video footage from on board the plane . robin 's comments follow claims by two magazines , german daily bild and french match . the video was recovered from a phone at the wreckage site . all 150 on board were killed .	Prosecutor denies reports of cell phone video from inside Germanwings crash plane.	French Prosecutor refuse reports of video from crash plane
https://edition.cnn.com/2015/04/01/opinions/shetty-end-executions/index.html	amnesty international releases its annual review of the death penalty worldwide ; much of it makes for grim reading . salil shetty : countries that use executions to deal with problems are on the wrong side of history .	55 people were found guilty of a range of offenses linked to violent attacks in the region and jailed . the public mass sentencing was part a china 's `` strike hard " campaign against unrest in xinjiang , a campaign the government claims was launched to combat `` terrorism	Humans do not deserve execution.	Human should not have capital punishment.

$$P_{final}(W)=\wp_{gen}P_{vocab}(W)+(1-\wp_{gen})\sum_{i:w_i=w}a_i$$

We use coverage mechanism to avoid repetition in which we use coverage vector C^t which is summation of attention distribution at'. We penalize the words that have already come by the use of attention distribution that see what has been covered so far.

$$C^t=\sum_{t'=0}^{t-1}a^{t'}$$

We also penalize the overlapping between attention distribution a^t and coverage vector C^t . So it does not cover anything that has been covered.

$$C^t=\sum_i min(a_i^t,c_i^t)$$

D. Encoder-Decoder Model

The Long Short term memory (LSTM) is a sequence to sequence model and is a Recurrent Neural Network. It takes a sequence as an input to encoder and generate another sequence as output from decoder. Fig 2 shows encoder-decoder model. This model contains 2 parts. The first part is LSTM encoder which encodes the input sequence. The other one is decoding LSTM which will generate an output sequence. To build our encoding layer, we are going to use a bidirectional RNN with LSTMs. Since we are using a bidirectional RNN, the outputs can be understood in 3 parts.

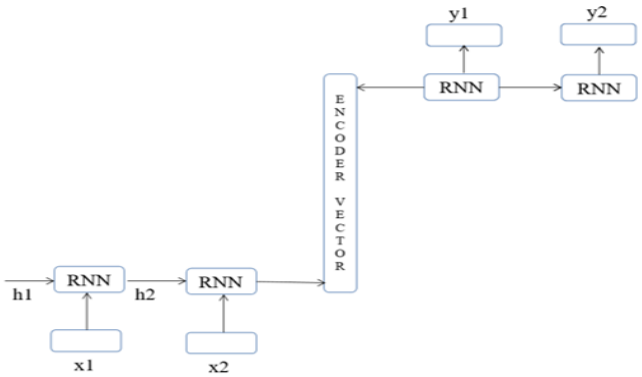

Fig. 3 Encoder-Decoder LSTM model

First one is Decoding cell which is just a two layer with dropout. Second One is attention in which we have used Bhadanau for attention style. Using this we can train our model fast as well as it will produce better results. Third one is getting our logits.

E. Generating Our Own Summaries

As an input we can either give our own descriptions or use them from the dataset. We just need to load few tensors to generate new summaries.

V. EVALUATION

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) set of matrices has been used to evaluate result. We have used recall and precision. Recall is a fraction which calculates how much of the reference summary is the system summary covering. Precision tells how much of the system summary is relevant. ROUGE-1 calculates overlapping of unigrams in machine generated summary and actual summary.

ROUGE-2 calculates overlapping of bigrams in machine generated summary and actual summary. ROUGE-L measures longest matching sequence of words using LCS.

We have used standard ROUGE metrics [5], we report F score for ROUGE 1 which measures the word overlap, ROUGE 2 which measures the bigram overlap and ROUGE L which measures the longest common sequence between the reference summary and the summary to be evaluated. Similar to [5] and [3] we have used pyrouge package to obtain our ROUGE scores given in table I.

	ROUGE 1	ROUGE 2	ROUGE L
F_score	0.3502	0.1381	0.3171
Recall	0.4006	0.1492	0.3111
Precision	0.3836	0.1588	0.3176

VI. RESULT

For Pointer Generator Model training we start with max_enc_steps=10, max_dec_steps=10 and interrupt at some times and increase the value of max_enc_steps and max_dec_steps to 400 and 100 respectively. For LSTM encoder decoder model we have prepared our own data set from generated summary.

We have used batched training of size 64, a learning rate of 0.005. We train our model on our prepared dataset from CNN. We have used 100 epochs which took 3 days. Results have been shown in Table II.

VII. CONCLUSION

In this paper, we have implemented both encoder-decoder LSTM model[5] and pointer generator model [4] with attention mechanism to generate headlines. We have trained our model using CNN/Dailymail stories and we have evaluated our result using manually. Most of the time the summary and headline is valid and grammatically correct. We have used bidirectional RNN to improve accuracy.

VIII. FUTURE WORK

Our model works fine for CNN/Dailymail stories. In future, we can extend our model to other domains to generate reliable summary. In this paper we have trained our model on CNN/Dailymails, in future it can be trained and tested on other domain specific datasets. Work can be done to improve the accuracy further.

REFERENCES

1. Lopyrev, K. (2015). Generating news headlines with recurrent neural networks. arXiv preprint arXiv:1512.01712.
2. Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. CoRR, abs/1509.00685, 2015.
3. See, A., Liu, P. J., & Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368.
4. DeepMind Q&A Dataset: https://cs.nyu.edu/~kcho/DMQA/ cited on:
5. Chin-Yew Lin. 2004b. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out: ACL workshop.
6. How to Configure the Learning Rate Hyperparameter When Training Deep Learning Neural Networks: https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/ (cited on: 22-May-2019)
7. Introduction toWord Vector: https://medium.com/@jayeshbahire/introduction-to-word-vectors-ea1d4e4b84bf (cited on: 22-May-2019)
8. Recurrent neural networks and LSTM https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5 (cited on: 22-May-2019)
9. Understanding Encoder-Decoder Sequence to Sequence Model https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346 cited on:
10. Python Language: https://www.geeksforgeeks.org/python-programming-language(cited on: 22-May-2019)
11. Machine learning course: https://www.coursera.org/learn/machine-learning
12. Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In Neural Information Processing Systems.
13. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In International Conference on Learning Representations.
14. Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In NIPS 2016 Workshop on Multi-class and Multi-label Learning in Extremely Large Label Spaces.
15. Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In Association for Computational Linguistics.
16. Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2016. Efficient summarization with read-again and copy mechanism. arXiv preprint arXiv:1611.03382 .
17. Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In Empirical Methods in Natural Language Processing.
18. Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Association for the Advancement of Artificial Intelligence.
19. Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. In Association for the Advancement of Artificial Intelligence.
20. Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Computational Natural Language Learning.
21. Philipp Koehn. 2009. Statistical machine translation. Cambridge University Press.
22. Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In Empirical Methods in Natural Language Processing.
23. Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In Association for Computational Linguistics.
24. Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning.
25. Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In International Joint Conference on Artificial Intelligence.

AUTHORS PROFILE

Anubha Agrawal is working toward the undergraduate degree in the Department of Computer Engineering, from Zakir Husain College of Engineering and Technology, AMU (Aligarh Muslim University), Aligarh, India. Her research interests include deep learning, text summarization, android development.

Sakshi Saraswat is pursuing her undergraduate degree from Computer Engineering Department, Zakir Hussain College of Engineering and Technology, Aligarh Muslim University, Aligarh. Her main interest lies in Machine Learning, Web Development, android Development, Deep Learning and Neural Networks.

Hira Javed works as an Assistant Professor in the Department of Computer Engineering AMU, Aligarh . Her current research interests include Data mining, Machine Learning and Natural Language Processing.