Text Summarization

Submitted by: Aya Yamin , Sojood Alsayyed
Submitted to: Dr. Hamed Adbelhaq[1]

[1] *An-Najah National University - Faculty of Graduate Studies*

May 27, 2022

Summarization task is refer to the task of creating a short summary that captures the main ideas of an input text. Therefore,in this work,we focus on abstractive summarization, which generates the summary through paraphrase ,so we model sequence-to-sequence Attentional Encoder-Decoder Recurrent Neural Networks recurrent neural networks model .

Decoder , Encoder , Attention

1 Introduction

NLP is a process of developing a system that can process and produce language as good as human can produce. The use of World Wide Web has increased and so the problem of information overload also has increased. Hence there is a need of a system that automatically retrieves, categorize and summarize the document as per users need. Document summarization is one possible solution to this problem. Text summarization is a process to express the content of a document in a condensed form that meets the needs of the user. More and more electronic data is available on the Internet and it is not possible to read everything and hence some form of information condensation is needed. Summarization serves as a tool which helps the user to efficiently find useful information from immense amount of information.

This paper is divided into six sections.Section 1: Introduction, Section 2 : System architecture and tools description, Section 3: Related work,Section 4:Overview of the system , Section 5: Results and Analysis, Section 6: Evaluation, Section 7:Conclusion.

2 System Architecture And Tools Description

Encoder Decoder RNN (Recurrent neural network) model is used in order to overcome all the limits faced by the NLP for text summarization such as getting a short and accurate summary. It is a much more intelligent and smart approach.

We can convert a document summmarization problem into a supervised and semi-supervised machine learning problem. Mostly we use sequence-2-sequence

architecture of RNN . A sequence2sequence model basically comprises of an Encoder and a Decoder connected sequentially so as to generate an output (Summary) for a given input (Text).

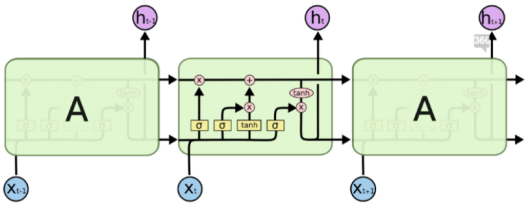
2.1 RNN for text summarization

In RNN, the new output is dependent on previous output. Due to this property of RNN we try to summarize our text as more human like as possible.
Training: Recurrent neural network use back propagation algorithm, but it is applied for every time stamp. It is commonly known as backpropagation through time(BTT).
We use a Sequence2Sequence model comprising of Encoders and Decoders to summarize our long input statements into a summarized line. These Encoders and decoders further comrpises of LSTM's as an extended functionality of a simple RNN.

2.2 LSTM's(Long-Short term memory network)

It is a special kind of RNN. Which is capable of learning long term dependencies thus treating the problem of short term dependencies of a simple RNN. It is not possible for a RNN to remember the to understand the context behind the input while we try to achieve this using a LSTM.

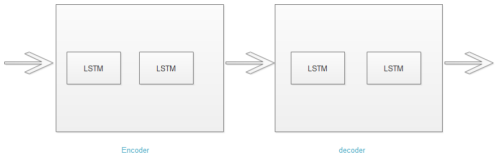
Here identify the information that is not



required and will be thrown away from cell state. This decision is made by sigmoid layer (as shown in figure) which is also known as forget layer. Using the next sigmoid layer we try to determine the new information that we are going to store. In this process, a sigmoid layer known as input gate layer decides which values will be updated. In the next step, we update the old cell state Ct-1 into Ct thus determining the new candidate values.

Ct=ft*Ct-1+it*Ct
In the last step we run a sigmoid layer which describes the output.
Ot=sigma(Wo[ht-1,xt]+bo)
where xt=New input,
bf=bias,
wf=weight,
ht-1=Output from previous time stamp
and ft=sigm(wf[ht-1,xt]+bf)

2.3 Encoder-Decoder Architecture



Encoder and Decoders are the two components of a sequence-2-sequence rnn architecture. These are majorly used when inputs and outputs are of varying lengths. In this architecture, various LSTMs are placed sequentially in order to encode and then decode the input.

Encoder - It encodes the entire input into a form suitable for processing.
Decoder - It decodes the processed input into an understable and required output.

One word at a time is provided as input to the encoder with its time stamp. This word is then processed in the LSTM by retieving the information present in the sequenced input. It is followed by a decoder that decodes the input sequence into a more readable and desirable output format. It is also dependent upon the time stamp. Finally, we produce a summary using the model which predicts the next relecant word by considering the previous sequence of words.

This model is an extension to feedforward network with atleast one feedback connection. In standard LSTM sequence of fixed length is passed as an input to be encoded into a fixed dimension vector (v), which is then decoded into the output sequence of words.

Advantages

- Generates more human like summary.
- Accuracy increases with the data size.

Disadvantages

- Complex model.
- Repetition of word and phases is not checked.
- Abstractive summarizers are comparitively slower.

3 Related Work

There are several papers that work on text summarization. One paper (Zhang, Xu, and Wang, 2019),In this paper, propose a novel pretraining-based encoder-decoder framework, which can generate the output sequence based on the input sequence in a two-stage manner. For the encoder of model, encode the input sequence into context representations using BERT. For the decoder, there are two stages in model, in the first stage, use a Transformer-based decoder to generate a draft output sequence. In the second stage, mask each word of the draft sequence and feed it to BERT, then by combining the input sequence and the draft representation generated by BERT, use a Transformer-based decoder to predict the refined word for each masked position. To the best of knowledge, the approach is the first method which applies the BERT into text generation tasks. As the first step in this direction, evaluate proposed method on the text summarization task. Experimental results show that model achieves new state-of-the-art on both CNN/Daily Mail and New York Times datasets.

Another paper work on extractive text summarization by selecting a subset of existing words, phrases or sentences from the original text to form summary.There used many of tools: 1- Statistical Approaches :can summarize a document using statistical features of the sentence like title, location, term frequency, assigning weights to the keywords and then calculating the score of the sentence and selecting the highest scored sentence into the summary. 2- Linguistic Approaches:Linguistic is a scientific study of language which includes study of semantics and pragmatics. Study of semantics means how meaning is inferred from words and concepts and study of pragmatics includes how meaning is inferred from context. (Munot and Govilkar, 2014)

The third paper which was reviewed (Rau, Jacobs, and Zernik, 1989) they discussed an The lack of extensive linguistic coverage is the major barrier to extracting useful information from large bodies of text. Current natural language processing (NLP) systems do not have rich enough lexicons to cover all the important words and phrases in extended texts. Two methods of overcoming this limitation are (1) to apply a text processing strategy that is tolerant of unknown words and gaps in linguistics knowledge, and (2) to acquire lexical information automatically from the texts.So this paper used two methods have been implemented in a prototype intelligent information retrieval system called SCISOR (System for Conceptual Information Summarization, Organization and Retrieval). This article describes the text processing, language acquisition, and summarization components of SCISOR.

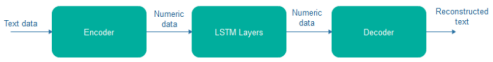
4 Overview of the system

4.1 Model summarization

For implementing our neural network we would use keras library of python along with other libraries required for data preprocess-

ing and designing. After loading the data we use the contraction mapping in order to deal with the contracted words of the language. It helps in better understanding(intended meaning) and improved accuracy of the model.

Moving on to the data preprocessing step, data cleaning is implemented along with removal of stopwords in order to make data ready for our model. This phase is implemented the same function for both summary and complete text of the training model.
In the next phase model building is implemented. 3 Layers of LSTM are added(can be increased or decreased accordingly) along with encoder and decoder.



It is followed by attention layer and a dense layer before the compilation phase. Finally we complete the summarization using the data generated and adding it sequentially using the decode seq method and seq2seq method.
It if followed by seq2text method to add the text into the sequence. In the end, we have our required summary.

4.2 Implementation

- Import the required libraries. Tensorflow and keras are the main libraries that we use in order to implement RNN along with other essential libraries.
- Read the data ,split abstract from text and preprocess the data.
  We use contraction mapping dictionary in order to map the contracted words with their intended meanings.

- Set the lengths for summary and text.

- Build the model. Key components of the model are as follows:

  Encoder inputs is used in order to encode the words into numeric data for processing by LSTM layers.
  We use 3 LSTM layers in order to process the data effectively. You can also experiment by adding or removing the layers in order to find more better accuracies.
  Decoder again converts the numeric data into the understandable word formats.

- Adding the Attention layer: Attention layer is used to selectively choose the relevant information while discarding the non-useful information by cognitively mapping the generated sentences with the inputs of encoder layer. Dense Layer: It mathematically represents the matrix vector multiplication in neurons and is used to change the dimensions of the vectors for processing between various layers.

- Decode the decoder sequence for text generation and convert the integer sequence into word sequence for summary.

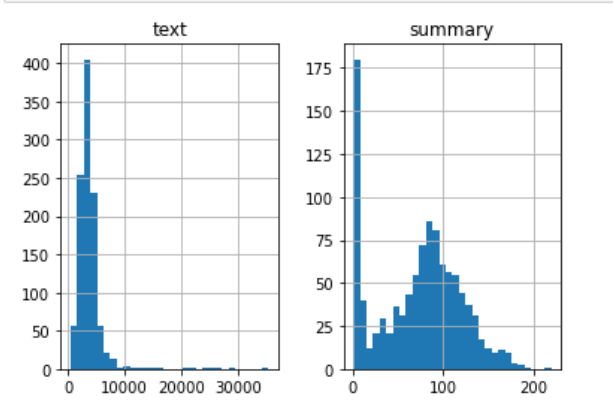
- predicted summary

- Evaluation accuracy and performance between original and predicted summary

5 Results and analysis

The first time we attempted to train model, we trained it for 5 epochs with batch size 64. We used 500- dimensional encoder and decoder cells. The loss decreased from 5.6770 to 1.0333, but upon viewing the predictions on the development set, we realized that the model had begun to output purely unknown tokens, for the maximum possible length output . To diagnose the problem, we revisited the data upon which we were training our

model.
Then, we increased the epochs to either 50 or 100 and we notice that the results improved,so we start changing maximum possible length output with drawing the distribution of input sequence .



Then, after different experiments ,we fixed maximum possible length output summary to 50 ,epochs to 50,batch size to 512 ,and train model on these parameters .
To train the model, we minimize the negative log probability of prediction word y given an input x and context yc.

$$loss_t = -log\ p(y_t \mid x, y_c)$$

$$CE(p, q) = -\sum_w q(w)\ log\ p(w) = -\ \log p(w)$$

6 Evaluation

Evaluations are done using different ways as shown below :

1. ROUGE (Recall Oriented Understudy for Gisting Evaluation) :
a set of metrics used for the evaluation of automatic text summarization and machine translations. The metrics basically compare automatically generated summary with reference summary or multiple reference summaries.

It includes following 3 evaluation metrics:

ROUGE-1: Measures overlap of unigram in reference summary and candidate (Machine generated) summary.

ROUGE-2: Measures content overlap of bi-grams.

ROUGE-L : Compute the Longest common subsequence between reference summary and candidate (Machine generated) summary. Each sentence in a summary is considered as a sequence of words. Two summaries which have longer common sequence of words are more similar to each other.

To evaluate how accurate our machine generated summaries are we compute the Precision, Recall and F-measure for any of this metric.
In ROUGE recall refers that how much words of candidate summary are extracted from reference summary. Formula to calculate recall:

$$R = \frac{Nnumber\ of\ overlaping\ words}{Total\ words\ in\ reference\ summary}$$

In ROUGE precision refers that how much candidate summary words are relevant. Formula to calculate precision:

$$P = \frac{Nnumber\ of\ overlaping\ words}{Total\ words\ in\ candidate\ summary}$$

F measure provides the complete information that recall and precision provides separately.

2. BLEU(Bilingual Evaluation Understudy): a metric widely used for models having a word sequence as output.The range of BLEU scores is between 0 and 1, where 0 signifies no match between the expected output and the predicted output and 1 means a perfect match.

BLEU can be considered as a modification to precision to handle sequence outputs. it clips the number of times a word is seen in the candidate or predicted output to the maximum times it appears in the reference or expected output.it also considers n-grams(bigrams, trigrams 4-grams).

7 Conclusion

Due to the information overload , strong text summarizers used to solve this problem.Hence there is a need to develop system where a user can efficiently retrieve and get a summarized document. One possible solution is to summarize a document using either extractive or abstractive methods. This paper focused on abstractive summarization .

References

Munot, Nikita and Sharvari S Govilkar (2014). "Comparative study of text summarization methods". In: *International Journal of Computer Applications* 102.12.

Rau, Lisa F, Paul S Jacobs, and Uri Zernik (1989). "Information extraction and text summarization using linguistic knowledge acquisition". In: *Information Processing & Management* 25.4, pp. 419–428.

Zhang, Haoyu, Jianjun Xu, and Ji Wang (2019). "Pretraining-based natural language generation for text summarization". In: *arXiv preprint arXiv:1902.09243*.